(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平11－134227

(43)公開日　平成11年(1999) 5月21日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　12/00	５１１	Ｇ０６Ｆ　12/00	５１１

審査請求　未請求　請求項の数28　ＯＬ　（全 21 頁）

(21)出願番号　特願平9－293765

(22)出願日　平成９年(1997)10月27日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　猪原　茂和
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者　鍵政　豊彦
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者　野田　文雄
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

(74)代理人　弁理士　小川　勝男

最終頁に続く

(54)【発明の名称】　ファイルフォーマット変換方法とこれを用いたファイルシステム及び情報システム及び電子商取引システム

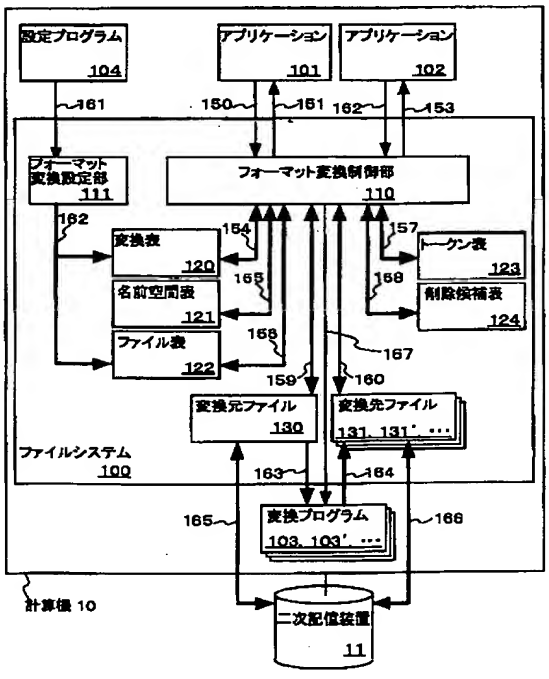
(57)【要約】

【課題】　複数のファイルフォーマットのファイルの間のフォーマット変換をユーザの関与なしに行う。

【解決手段】　ファイルシステム１００が変換元ファイル１３０と変換先ファイル１３１の関係を保持し、ファイル操作ＡＰＩの発行に同期してフォーマット変換に関する処理を行う。

【効果】　ユーザはアプリケーション上で行うユーザの本来の作業を行えばよくなり、その過程で必要となる様々なフォーマット変換（１段階でも、多段階でも良い）はユーザの関与なしで行われる。この際、変換元ファイルを指定したり、フォーマット変換のタイミングを指定する必要がなく、ユーザは常に最新の変換先ファイルを利用可能となる。

図1



【特許請求の範囲】

【請求項１】所定のフォーマットを有する第１のファイルを変換プログラムにより該フォーマットと異なるフォーマットを有する第２のファイルに変換するファイルフォーマット変換方法であって、（１）ユーザのアプリケーションにより第２のファイルの１つを指定し、（２）第１のファイルと、該第１のファイルから該指定した第２のファイルを得る変換プログラムとを対応づけ、（３）該変換プログラムにより第１のファイルから少なくとも１つの所望の第２のファイルへのフォーマット変換を行う、ファイルフォーマット変換方法。

【請求項２】前記（１）の指定する処理は、前記第２のファイルの１つに対してユーザのアプリケーションが発行するファイル操作を用いてなされる請求項１記載のファイルフォーマット変換方法。

【請求項３】第１のフォーマットを有する第１のファイルを変換プログラムP_1，P_2，…P_m（m：１以上の整数）によりそれぞれ第１のフォーマットと異なる第２のフォーマットを有する第２のファイルF1，F_2，…F_mに変換するファイルフォーマット変換方法であって、（１）第１のファイルと変換プログラムとそれぞれの変換プログラムにより変換される第２のファイルとを対応づけ、（２）ユーザのアプリケーションにより第１のファイル又は第２のファイルを指定し、（３）①ユーザのアプリケーションが発行する第１のファイル又は第２のファイルへのファイル操作の時刻、②所定の時刻、③所定の条件が成立した時刻、のうちの少なくとも１つの時刻を契機として、上記対応づけられた変換プログラムにより第１のファイルから第２のファイルへのフォーマット変換を行う、ファイルフォーマット変換方法。

【請求項４】第１のフォーマットを有する第１のファイルを変換プログラムP_1，P_2，…P_m（m：１以上の整数）によりそれぞれ第１のフォーマットと異なる第２のフォーマットを有する第２のファイルF_1，F_2，…F_mに変換するファイルフォーマット変換方法であって、（１）第１のファイルと変換プログラムとそれぞれの変換プログラムにより変換される第２のファイルとを対応づけ、（２）上記変換プログラムはそれぞれ上記第１のファイルから第３のファイルを得る第１の中間変換プログラムと、第３のファイルから上記第２のファイルを得る第２の中間変換プログラムとからなり、（３）ユーザのアプリケーションにより第１のファイル又は第２のファイルを指定し、（４）アプリケーションが発行する第１のファイルへのファイル操作を第１の契機として上記対応する変換プログラムの第１の中間変換プログラムにより第１のファイルから第３のファイルを得、該第３のファイルへのファイル操作を第２の契機として第２の中間変換プログラムにより第３のファイルから第１のファイルを得る、ファイルフォーマット変換方法。

【請求項５】前記第１の契機として、第１のファイルへの書き込み操作または書き込みのためのオープン操作を用い、前記第２の契機として第２のファイルへの読み出し操作または読み出しのためのクローズ操作を用いる請求項４記載のフォーマット変換方法。

【請求項６】前記第１のファイルが第１の計算機に格納され、前記第２のファイルと前記第３のファイルが第２の計算機に格納される請求項４記載のフォーマット変換方法。

【請求項７】第１のフォーマットを有する第１のファイルを変換プログラムP_1、P_2、…P_m（m：１以上の整数）によりそれぞれ第１のフォーマットと異なる第２のフォーマットを有する第２のファイルF_1、F_2、…F_mに変換するファイルフォーマット変換方法であって、（１）第１のファイルと変換プログラムとそれぞれの変換プログラムにより変換される第２のファイルとを対応づけ、（２）第１のファイルのファイル名から第２のファイルのファイル名を生成し、（３）生成したファイル名で第２のファイルをユーザに提供し、（４）提供されたユーザのアプリケーションにより該ファイル名で第１のファイル又は第２のファイルを指定し、（５）アプリケーションが発行する第１のファイル又は第２のファイルへのファイル操作を契機として上記対応づけられた変換プログラムにより第１のファイルから第２のファイルへのフォーマット変換を行う、ファイルフォーマット変換方法。

【請求項８】前記（２）のファイル名を生成する処理は、前記第１のファイルの作成操作を契機として行う請求項７記載のファイルフォーマット変換方法。

【請求項９】前記（２）のファイル名を生成する処理は、第２のファイルのファイル名を含むディレクトリに対する検索操作または表示操作を契機として行う請求項７記載のファイルフォーマット変換方法。

【請求項１０】前記（１）の対応づける処理は、第１のファイルと、第１の変換プログラムのうちの一つと、当該変換プログラムにより第１のファイルから得られる第２のファイルの３種のうち、第１のファイルから対応する２種を得る方法、または、第２のファイルから対応する２種を得る方法を含む請求項３乃至９いずれか１項記載のファイルフォーマット変換方法。

【請求項１１】前記対応づけの方法として、第１のファイルと、第１の変換プログラムと、第２のファイルとの間の対応関係を保持する表を用いる請求項１０記載のファイルフォーマット変換方法。

【請求項１２】前記対応づけの方法として、第１のファイルと、第１の変換プログラムと、第２のファイルとの間の対応関係を保持するプログラムを用いる請求項１０記載のファイルフォーマット変換方法。

【請求項１３】前記ファイル操作は、第１のファイルへの書き込み操作か書き込み後のクローズ操作、または第２のファイルへの読み出し・書き込み操作か読み出し・

書き込みのためのオープン操作である、請求項３、４、６、７いずれか１項記載のファイルフォーマット変換方法。

【請求項１４】前記第１及び第２のファイルと異なる第３のファイルがあり、
前記第１のファイルから第２のファイルへのフォーマット変換時に、前記第３のファイルから第１のファイルへのフォーマット変換を行う請求項１乃至１３項いずれか１項記載のファイルフォーマット変換方法。

【請求項１５】前記第１のファイルまたは前記第２のファイルのいずれかが、電子メール、または電子メールの添付ファイルである請求項１乃至１４いずれか１項記載のファイルフォーマット変換方法。

【請求項１６】第１のファイルと、第１のファイルをフォーマット変換して得られる１つ以上の第２のファイルをユーザに提供するファイルフォーマット変換方法であって、第１のファイルに対する第１のファイル操作と、第２のファイルの１つである第３のファイルに対する第２のファイル操作のうち、同時に実行する操作をたかだか１つとする、ファイルフォーマット変換方法。

【請求項１７】前記第１のファイル操作と前記第２のファイル操作のどちらかが、書き込み操作、または書き込みモードのオープン操作である請求項１６記載のファイルフォーマット変換方法。

【請求項１８】フォーマット変換によって得た前記第２のファイルの一部または全部について、ファイル名を削除せずにファイルの内容を削除する請求項１乃至１７いずれか１項記載のファイルフォーマット変換方法。

【請求項１９】請求項１－１８いずれか１項記載のファイルフォーマット変換方法をライブラリとして記録した記録媒体。

【請求項２０】複数のファイルを保持する二次記憶装置を具備した計算機システムに搭載されるファイルシステムであって、
変換元ファイルと、フォーマット変換後の変換先ファイルと、変換元ファイルと変換プログラムと変換先ファイルとを対応づける対応指示部と、アプリケーションプログラムが行い得るファイル操作を規定するアプリケーション・プログラミング・インターフェース（以下、ＡＰＩ）と、該ＡＰＩの起動に応じて所望のフォーマット変換を実行するフォーマット変換制御部とからなる、ファイルシステム。

【請求項２１】複数のファイルを保持する二次記憶装置を具備した計算機システムが複数個ネットワークを介して接続されている分散システムにおける各計算機システムに搭載されるファイルシステムであって、
変換元ファイルとフォーマット変換後の変換先ファイルとの少なくとも一方からなるファイルと、変換元ファイルと変換プログラムと変換先ファイルとを対応づける対応指示部と、アプリケーションプログラムが行い得るファイル操作を規定するアプリケーション・プログラミング・インターフェース（以下、ＡＰＩ）と、ネットワークを介して他のファイルシステムと相互に通信を行いながら該ＡＰＩの起動に応じて所望のフォーマット変換を実行するフォーマット変換制御部とからなる、ファイルシステム。

【請求項２２】前記対応指示部は、対応づけを登録または削除するインターフェースを含む請求項２０または２１項記載のファイルシステム。

【請求項２３】ファイル編集の機能を有し電源のオン／オフが起こりうる第１のパーソナルコンピュータ（以下、ＰＣ）と、電源がオン状態になっているＷｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ（以下、WWW）サーバ計算機と、ファイル変換の機能を有しない第２のパーソナルコンピュータ（以下、ＴＰＣ）とをネットワークで結合した情報システムであって、
ＰＣは、所定のフォーマットを有する変換元ファイルと、アプリケーションプログラムが行いうるファイル操作を規定するアプリケーション・プログラミング・インターフェース（以下、ＡＰＩ）を通したユーザからの指示に基づき内蔵の変換プログラムによるフォーマット変換を制御する制御部とを備え、
WWWサーバ計算機は、ＰＣにおけるフォーマット変換で得られた中間ファイルと、該中間ファイルを少なくとも１個の変換プログラムによりフォーマット変換して得られた少なくとも１個の変換先ファイルと、ＡＰＩを通したユーザからの指示に基づき該フォーマット変換を制御するフォーマット変換制御部とを備え、
ＴＰＣは、WWWサーバ計算機中にある上記変換先ファイルの少なくとも１個をWWWブラウザ経由で指定してその内容の読み出しをWWWサーバ計算機に対して指示する機能を有する、情報システム。

【請求項２４】複数のファイルを保持する二次記憶装置を備えたパーソナル・コンピュータであって、該パーソナル・コンピュータは、ファイル編集の機能を備えるアプリケーションと、１つ以上のフォーマットのうちのあるフォーマットを有する文書の表示プログラムと、変換元ファイルを変換先ファイルにフォーマット変換するために請求項１－７いずれか１項に記載のファイルフォーマット変換方法を有するファイルシステムとを備え、前記アプリケーションは、ユーザの操作に従って変換元ファイルを作成して二次記憶装置に保存し、表示プログラムは、オープンＡＰＩを発行し、該発行に同期して前記ファイルシステムにおいて変換元ファイルから変換された他の変換先ファイルの内容を読み出して表示する、パーソナル・コンピュータ。

【請求項２５】複数のファイルを保持する二次記憶装置を備えたサーバ計算機であって、該サーバ計算機は、ファイル編集機能を有するアプリケーションと、１つ以上のフォーマットのうちのあるフォーマットを有する文書

を表示または編集するプログラムと、変換元ファイルを変換先ファイルにフォーマット変換するために請求項１－１５いずれか１項記載のファイルフォーマット変換方法を有するファイルシステムとを備え、前記アプリケーションは、ユーザの操作に従って変換元ファイルを作成して二次記憶装置に保存し、WWWサーバは、機能限定パーソナル・コンピュータ（以下、ＰＣという）とネットワークで結合され、該機能限定ＰＣに搭載された前記プログラムからのファイル読み出し要求を受けてオープンＡＰＩ（アプリケーション・プログラミング・インターフェース）を発行し、該発行に同期して前記ファイルシステムにおいて前記変換元ファイルから変換された変換先ファイルの内容を読み出して前記プログラムに返送し、分散ファイル・サーバは、変換先ファイルフォーマットの１つを有する文書を表示または編集するプログラムに前記ネットワーク経由で結合され、該プログラムから変換先ファイルの読み出し要求を受けてオープンＡＰＩを発行し、該発行に同期して前記ファイルシステムにおいて変換元ファイルから変換された変換先ファイルの内容を読み出して該プログラムに返送する、サーバ計算機。

【請求項２６】複数のファイルを保持する二次記憶装置を備えた検索サーバ計算機であって、WWWクライアントと、少なくとも１つの検索サーバと、変換元ファイルを変換先ファイルにフォーマット変換するために請求項１－１５いずれか１項記載のファイルフォーマット変換方法を有するファイルシステムとを備え、WWWクライアントは、それぞれ異なるフォーマットを有するファイルを保持した複数のWWWサーバ計算機とネットワークで結合され、各ファイルの内容を収集してそれぞれ変換元ファイルとして前記ファイルシステムに格納し、検索サーバは、検索すべき情報の種類対応に設けられ、前記ファイルシステムにおいて異なるフォーマットを有する変換元ファイルをフォーマット変換した後の同一フォーマットを有する変換先ファイル群に対して前記情報の種類に対応した検索サーバによる検索を行う、検索サーバ計算機。

【請求項２７】前記検索サーバは、第１のフォーマットを入力として受け取る第１の検索プログラムと第２のフォーマットを入力として受け取る第２の検索プログラムとを搭載し、前記ファイルシステムにより前記情報を第１のフォーマットと第２のフォーマットにフォーマット変換して、第１の検索プログラムと第２の検索プログラムに提供する請求項２６記載の検索サーバ計算機。

【請求項２８】複数のファイルを保持する二次記憶装置を備え、暗号化機能を有するサーバ計算機と、復号化機能を有するクライアント計算機とが通信ネットワークで結合された電子商取引システムであって、サーバ計算機は、ファイルを編集する機能を有する編集プログラムと、暗号化プログラムと、該暗号化プログラムを変換プログラムとして請求項１－１５いずれか１項記載のファイルフォーマット変換方法により変換元ファイルを変換先ファイルにフォーマット変換するファイルシステムとを備え、前記編集プログラムより入力された商取引情報を該ファイルシステムにおける変換元ファイルに格納し、該変換元ファイルを前記暗号化プログラムにより暗号化して変換先ファイルに格納し、該変換先ファイルを通信ネットワークを経由してクライアント計算機に送信するものであり、クライアント計算機は、前記編集プログラムと、復号化プログラムと、該復号化プログラムを変換プログラムとして請求項１－１５いずれか１項記載のファイルフォーマット変換方法により変換元ファイルを変換先ファイルにフォーマット変換するファイルシステムとを備え、該ファイルシステムにおける変換元ファイルに前記送信されたファイルの内容を格納し復号化プログラムにより該変換元ファイルを復号化し、復号化した変換元ファイルを前記ファイルシステムにおけるファイルフォーマット変換方法により変換先ファイルにフォーマット変換し、前記編集プログラムにより該変換先ファイルの内容を参照する、電子商取引システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、計算機システムに関し、特に、ユーザに対して複数のファイルフォーマットで情報を提供するファイルシステムのフォーマット変換方法に関し、さらに具体的には、Ｗｏｒｌｄ―Ｗｉｄｅ　Ｗｅｂ（以下、WWWという）によって複数の計算機が複数のファイルフォーマットで情報交換を行う際に好適なファイルフォーマット変換方法とこれを用いたファイルシステム及び種々の情報処理システムに関する。

【０００２】

【従来の技術】従来の技術を説明する前に、まず、いくつかの用語を説明する。

【０００３】計算機システムにおいては、計算機の電源がオフになっても保持されるべきデータを格納するために「二次記憶装置」が用いられる。現在用いられている二次記憶装置の例としては、ハードディスク、フロッピーディスク、光磁気ディスクなどがある。

【０００４】「ファイルシステム」は、二次記憶装置とユーザの応用プログラム（「アプリケーション」）の間のデータの交換を司るソフトウェアである。ファイルシステムが二次記憶装置のデータを操作する単位は「ファイル」と呼ばれる。一般に複数のファイルがひとつの二次記憶装置に保持され、それぞれのファイルは「ファイル名」で区別される。ファイル名には文字列がよく使用される。ファイルシステムは多くの場合オペレーティング・システム（ＯＳ）の一部としてユーザに提供されるが、ライブラリとＯＳの組み合わせ、あるいはライブラリのみで提供される場合もある。すなわち、例えばＯＳが提供するファイルシステムと、当該ファイルシステム

に拡張機能を付加するライブラリとを組み合わせたソフトウェアも、二次記憶装置とアプリケーションの間のデータの交換を司るソフトウェアであればファイルシステムと呼ぶ。

【０００５】応用プログラム（アプリケーション）が行なうことのできるファイルへの操作は、アプリケーションとファイルシステムとのアプリケーション・プログラミング・インターフェース（ＡＰＩ）によって定義されている。特に、ファイルの入出力に関するＡＰＩを「ファイル入出力ＡＰＩ」と呼ぶ。ファイル入出力ＡＰＩには、オープン（ファイル読み書きの準備）、読み出し（ファイルからアプリケーションへのデータの移動）、書き込み（アプリケーションからファイルへのデータの移動）、クローズ（ファイル読み書きの終了）が含まれる。また、ファイルとディレクトリの操作に関するＡＰＩを「ファイル管理ＡＰＩ」と呼ぶ。ファイル管理ＡＰＩには、ファイル作成、ファイル名変更、ファイル削除が含まれる。ファイル入出力ＡＰＩとファイル管理ＡＰＩで可能となるファイルへの操作を総称して「ファイル操作」と呼ぶ。

【０００６】ファイル内のデータが持つ構造を「ファイルフォーマット」（または単に「フォーマット」）と呼ぶ。フォーマットの例としては、「改行文字で区切られた文字列の並び」、「タブ文字で区切られた項目群の並び」、「ワープロソフトＡのファイル形式」、「動画のコマ（ある瞬間に表示される画像）の並び」などがある。ファイル名の最後の数文字を用いて、ファイルフォーマットを表現することがある。このファイル名の最後の数文字を「拡張子」と呼ぶ。

【０００７】以下、従来の技術について述べる。

【０００８】現在多くのアプリケーション（例えばワード・プロセッサ、表計算、スケジュール管理、電子メール、プログラミング・ツールなど）が用いるファイルは、それぞれ固有のフォーマットを持っている。特に、近年広く使われるようになったパーソナル・コンピュータでは、多くのファイルフォーマットが使われており、その数は２００近くにものぼる。また、インターネットの発展に伴って出現したＷｏｒｌｄ―Ｗｉｄｅ　Ｗｅｂ（WWW）では、単なる文字だけでなく、静止画像、動画像、音声、コンピュータ・グラフィクスなど、多種多様なフォーマットが使用されている。

【０００９】一般にすべてのアプリケーションがすべてのファイルフォーマットをアクセス可能ではない。このため、ユーザがある情報をあるファイルフォーマットでファイルに格納したとしても、別のアプリケーションからその情報にアクセスするために、そのファイルを「フォーマット変換」を行う必要が生じることが多い。フォーマット変換の入力となるファイルを「変換元ファイル」、出力となるファイルを「変換先ファイル」と呼ぶ。

【００１０】フォーマット変換には、多種多様なフォーマットに関する多数の「変換プログラム」から、（１）ユーザの所望する変換を行う変換プログラム（または複数の変換プログラムの組合わせ）を選ぶ作業と、（２）選んだ変換プログラムに対して、変換元ファイルと変換先ファイルとを正しく与え、適切な時点で変換プログラムを実行する作業が必要となる。これらの作業は、アプリケーション上で行われるユーザの本来の作業とは関係ない。このため、フォーマット変換に関するユーザの負担をできる限り小さくすることが望まれる。特に上述のようにインターネットとWWWの利用が激増していることから、多種多様なフォーマットをできる限り簡単に扱える必要性が高まっている。

【００１１】これまでに主に上記（１）の作業を簡単化するために、以下の方法が提案されている。

【００１２】特開平６―１８７２１９号「プログラム間の自動データ共有方法」（以下公知例１と記す）では、利用するアプリケーションとファイルの名前をユーザが指示し、アプリケーション・データ対応表を検索して適切な変換プログラムを選択し、当該アプリケーションで当該ファイルを利用するために必要なフォーマット変換を行う。

【００１３】これにより、ユーザは上記（１）の作業、すなわち当該ファイルをアクセスするために使用すべき変換プログラムを選ぶ作業から解放される。一方（２）の作業、すなわちフォーマット変換をいつどのファイルに対して行うかを指定する作業は、依然としてユーザに課せられたままである。なぜなら、公知例１のユーザは、アプリケーション上で行われるユーザの本来の作業には必要ない、変換したいファイルをアプリケーション名とともに公知例１のシステムに与えるという作業を要求されるためである。

【００１４】また、WWWを含む現在の計算機システムでは、１つのアプリケーションが複数のファイルフォーマットを取り扱い、また１つのファイルフォーマットを取り扱うアプリケーションが複数存在する。このような状況ではアプリケーションとファイルの名前を指定しても変換方法が一意に定まらない。例えば、．ｔｅｘ、．ｐｓ、．ｐｄｆの３つの拡張子によって区別される３つのファイルフォーマットがあり、プログラムＡが．ｐｓと．ｐｄｆを取り扱える場合を考える。ユーザがプログラムＡとファイルｆｏｏ．ｔｅｘを指定しても、ｆｏｏ．ｔｅｘをｆｏｏ．ｐｓに変換すべきか、ｆｏｏ．ｐｄｆに変換すべきかは一意に決まらない。

【００１５】特開平９―６９０５９号「ファイル形態変換システム」（以下公知例２と記す）では、変換元ファイル名と変換先ファイル名（または変換元ファイルを使用するアプリケーション名と変換先ファイルを使用するアプリケーション名）をユーザが指定すると、変換プログラムを選びまたは組み合わせて、変換元ファイルから

変換先ファイルを得るフォーマット変換を行う。これにより、ユーザは上記（1）の作業、すなわち当該ファイルをアクセスするために使用すべき変換プログラムを選びまたは組み合わせる作業から解放される。一方（2）の作業、すなわちフォーマット変換をいつどのファイルに対して行うかの指定は、ユーザが行う必要があり、この負担はユーザに課せられたままである。

【0016】また複雑なファイル処理過程を一括して自動化する方法としては、University of California, Computer SystemsResearch Group著の文献「4.4BSD User's Reference Manual」のMAKE（1）章（O'Reilly&Associates, Inc, 1994）に記載のソフトウェア（以下公知例3と記す）が知られている。公知例3はソース・プログラムからバイナリ・プログラムへの、一段階または多段階にわたるコンパイル作業を簡単化する方法である。公知例3も（1）の課題を解決するが、（2）の課題は解決しない。

【0017】

【発明が解決しようとする課題】上述のように、多くの提案にもかかわらず従来のフォーマット変換方法では、上記（2）の作業がユーザに与える負担に対する配慮がされていなかった。（2）の作業はさらに以下の2つの作業に分けることができる。

【0018】（2—a）どのファイルを変換元ファイルとし、どのファイルを変換先ファイルにするかを指定する作業。この指定を間違えると、変換先ファイルの内容がユーザ所望のものにならなかったり、他のファイルの内容を破壊する危険性がある。特にフォーマット変換以前には変換先ファイルが存在しない場合がある点に注意が必要である。

【0019】（2—b）フォーマット変換をいつ行うかを指定する作業。この指定を間違えると、アプリケーションが古い情報にアクセスする危険性がある。

【0020】加えて、従来のフォーマット変換方法では、以下の課題があった。

【0021】（i）変換元ファイルと変換先ファイルの間の一貫性をとる方法が用意されていなかったため、変換元ファイルと変換先ファイルにほぼ同時に書き込みがおこって、続くフォーマット変換によってどちらかの書き込みが失われる、などの危険性がある。

【0022】（ii）多数の変換先ファイルを保持するために、二次記憶領域が余分に必要となる。

【0023】（iii）変換元ファイルが何らかの理由（例えば変換元ファイルをあるアプリケーションが編集中だったり、変換元ファイルを保持する二次記憶装置の電源がオフになっている等）で利用不能な間は、フォーマット変換を行うことができない。

【0024】従って本発明の目的は、複数のファイルフォーマットが複数のアプリケーションから利用可能な現状を考慮した上で、上記の（1）とともに、（2—a）、（2—b）の作業にともなう課題に加えて上記（i）、（ii）、（iii）の課題を解決し、複数のファイルフォーマットを利用する際のユーザの手間をできる限り減らすことである。

【0025】

【課題を解決するための手段】（2—a）の作業に伴う課題は、変換元ファイルと変換先ファイルがそれぞれ多数あり、かつこれらの間の関係があらかじめ与えられていないことに起因している。このために、どのファイルをどのファイルへフォーマット変換するかをユーザがその都度指示する必要があったが、本発明では、変換元ファイルと変換先ファイルと変換プログラムとの間の対応関係（1段階でも、多段階でも良い）をファイルシステム中に保持する。この手段によって、フォーマット変換方法が変換先ファイルのみから変換元ファイルと変換プログラムとを得ることを可能とした。同時に、変換元ファイルのみから1つ以上の変換先ファイルと、対応する1つ以上の変換プログラムを得ることも可能とした。

【0026】また、変換先ファイルが変換以前には存在しない場合に対応するために、変換元ファイルのファイル名から変換先ファイルのファイル名を得るファイル名変換方法をファイルシステム中に保持する。通常なら変換先ファイルはフォーマット変換の結果としてできるが、本発明では変換先ファイルを指定することによりフォーマット変換を開始することができるようにしてユーザの手間を減らす。すなわちフォーマット変換と変換先ファイルがいわゆる「鶏と卵」の状態になる。この問題を、ファイル名変換方法を備えることにより解決し、フォーマット変換を行う以前に変換先ファイルをユーザに提供することを可能にした。

【0027】（2—b）は、アプリケーションとフォーマット変換方法の連携をとる手段がないこと、すなわちアプリケーションがどのファイルをいつ、どのように操作するかをフォーマット変換方法が知る術がなかったことに起因する。このため従来は、フォーマット変換の起動作業を、アプリケーションによる作業とは別にユーザが行わざるを得なかった。これに対し本発明では、ユーザによるファイル入出力ＡＰＩの発行を契機として、ファイルシステムから変換プログラムを起動して実行する。

【0028】以上の2点を備えることにより、本発明のフォーマット変換方法は、アプリケーションがどのファイルを操作の対象とし、アクセスが書き込みか読み出しかを知ることができる。すなわち、ユーザはアプリケーション上で行うユーザの本来の作業を行えばよくなり、その過程で必要となる様々なフォーマット変換（1段階でも、多段階でも良い）はユーザの関与なしで行われる。特にファイル操作を契機として変換プログラムを実

行するため、フォーマット変換のタイミングを指定する必要がなく、ユーザは常に最新の変換先ファイルが利用可能となる。

【００２９】加えて本発明では、上記（ｉ）、（ｉｉ）、（ｉｉｉ）の開題を解決するため、それぞれ以下（Ｉ）、（ＩＩ）、（ＩＩＩ）の手段を設ける。

【００３０】（Ｉ）変換元ファイルに対するファイル入出力ＡＰＩの実行と、変換先ファイルに対するファイル入出力ＡＰＩの実行との排他制御を行う。すなわち、いずれか一方が実行されている間は他方を実行しないよう制御する。これにより、変換元ファイルと変換先ファイルの一貫性を保持できる。

【００３１】（ＩＩ）多数の変換先ファイルを保持することによる二次記憶領域の無駄を解決するために、変換先ファイルを必要に応じて消去する。

【００３２】（ＩＩＩ）変換元ファイルが利用不能な間もフォーマット変換を行うことを可能にするために、中間ファイルを導入し、変換元ファイルから中間ファイルへ、中間ファイルから変換先ファイルへの２段階の変換を行う。

【００３３】ファイルシステムはファイルに対するあらゆる操作要求を受け取るので、ファイル入出力ＡＰＩを契機とした変換や、排他制御、二次記憶領域の無駄の管理、中間ファイルの管理が実現できる。さらにファイルシステムは、多くのアプリケーションによって共有されているので、上記解決手段をファイルシステムが備えることによって多くのアプリケーションがこの発明の利益を享受することができる。

【００３４】

【発明の実施の形態】以下、本発明の実施の一形態を、図面を参照しながら説明する。なお簡単のため、本明細書中では以下に述べる発明の実施の形態を「第１の実施例」と呼ぶことにする。

【００３５】全体構成

図１と図２を用いて第１の実施例の構成を説明する。なお、図１において、矢印付きの太い線（１５４―１６０、１６２―１６６）は主要なデータの流れを、また矢印付きの細い線（１５０―１５３、１６１、１６７）は主要な制御の流れを表わす。

【００３６】計算機１０は、いわゆるパーソナル・コンピュータ、ワークステーション、並列計算機、大型計算機等、任意のコンピュータでよい。二次記憶装置１１はファイルを保存する記憶装置である。二次記憶装置１１には不揮発性記憶装置（磁気ハードディスク、光ディスク等）がよく用いられるが、特別な場合には揮発性記憶装置（メインメモリ、キャッシュメモリ等）が用いられる場合もある。計算機１０と二次記憶装置１１の間の結合には様々な形態が考えられるが、計算機１０にファイルを供給できればよく、特定の結合形態を必要としない。例えば、計算機１０が占有する通信線による結合、複数の計算機によって共有されるネットワークを介した結合、他の計算機経由の結合等が代表的な例として考えられる。また、計算機１０に対して二次記憶装置１１を１つ以上用いても差し支えない。

【００３７】ファイルシステム１００は計算機１０上で動作するソフトウェアである。ファイルシステム１００はデータをファイルと呼ぶ単位で管理、保存する。１つ以上のファイルが二次記憶装置１１に保持され、それぞれのファイルはファイル名で区別される。

【００３８】アプリケーション１０１、１０２はユーザの実行する応用プログラムである。多くのアプリケーションはファイルを操作するためにファイルシステム１００とのやり取りを行う。アプリケーション１０１、１０２がファイルシステム１００に対して要求するファイル操作は、ファイル入出力ＡＰＩとファイル管理ＡＰＩを含む。

【００３９】変換元ファイル１３０は、フォーマット変換の入力となるファイルである。変換先ファイル群１３１、１３１’、…は、フォーマット変換の出力となるファイルである。なお、第１の実施例では説明を明瞭にするために変換元ファイル１３０と変換先ファイル群１３１、１３１’、…を別々のファイルにしてあるが、１つのファイルが変換元ファイル１３０かつ変換先ファイル１３１であっても差し支えない。また図１に示した変換元ファイルと変換先ファイルは、それぞれ複数存在して差し支えない。

【００４０】変換プログラム群１０３、１０３’、…はファイルシステム１００が実行するフォーマット変換のためのプログラム群である。本実施例では、変換プログラム群１０３、１０３’、…はファイルシステム１００の外に置かれてフォーマット変換を実行しているが、ファイルシステム１００に内蔵して、ファイルシステム１００内でフォーマット変換を実行しても差し支えない。

【００４１】変換表１２０は、変換元ファイルと変換先ファイルの組を、変換プログラムに対応づける表である。図２に示すように、変換表１２０は１つ以上の変換表エントリ２００からなり、１つの変換表エントリ２００が１通りのフォーマット変換に対応する。変換表エントリ２００はさらに、変換元フォーマット２０１、変換先フォーマット２０２、変換プログラム２０３を含む。変換元フォーマット２０１は変換元ファイルのフォーマット、変換先フォーマット２０２は変換先ファイルのフォーマットである。変換プログラム２０３は変換元フォーマット２０１のフォーマットを持つ変換元ファイルから、変換先フォーマット２０２のフォーマットを持つ変換先ファイルへの変換を行うプログラムで、その名前（および必要ならば起動時引数）を持つ。

【００４２】名前空間表１２１は、ファイルシステム１００が管理するファイル群のファイル名と、ファイルシステム内部で利用されるファイルＩＤを対応づける。図

２に示すように、名前空間表１２１は１つ以上の名前空間表エントリ２１０からなり、１つの名前空間表エントリ２１０が１つのファイルに対応する。名前空間表エントリ２１０はさらに、ファイル名２１１とファイルＩＤ２１２を含む。ファイル名２１１は名前空間表エントリ２１０のファイルのファイル名、ファイルＩＤ２１２は当該ファイルのファイルＩＤである。ファイルＩＤは、ファイルシステム１００内部で使用されるファイルの番号で、ファイルＩＤとファイルは一対一に対応する。

【００４３】ファイル表１２２は、ファイルシステム１００が管理するファイルの諸属性を保持する。特に、ファイルのフォーマットと、（もし当該ファイルが変換先ファイルであれば）変換元ファイルをこの表に保持する。図２に示すように、ファイル表１２２は１つ以上のファイル表エントリ２２０からなり、１つのファイル表エントリ２２０が１つのファイルに対応する。ファイル表エントリ２２０はさらに、ファイルＩＤ２２１、フォーマット２２２、変換元ファイルＩＤ２２４、トークンＩＤ２２５、ファイル内容２２６を含む。ファイルＩＤ２２１は当該ファイルのファイルＩＤ、フォーマット２２２は当該ファイルのファイルフォーマット、タイムスタンプ２２３は当該ファイルに最後に書き込みがあった時刻、変換元ファイルＩＤ２２４は（もし当該ファイルが変換先ファイルであれば）変換元ファイルのファイルＩＤ、トークンＩＤ２２５は当該ファイルに対して割り当てられトークン表１２３で管理されるトークンＩＤである。ファイル内容２２６は当該ファイル本体（すなわちファイルのデータを格納する部分である。図１の変換元ファイル１３０や変換先ファイル群１３１、１３１’、…）および当該ファイルの属性である。

【００４４】なお、変換元フォーマット２０１、変換先フォーマット２０２、フォーマット２２２に格納すべきフォーマットの表現方法にはいくつか考えられるが、本発明はこの方法には依存しない。例えば、拡張子がファイルのフォーマットを表わしている計算機システムでは、拡張子を格納することができる。また、ファイル内容２２６の一部または全部によってフォーマットが決定される場合もあるが、この場合にはフォーマット毎に異なるフォーマット名を決めておき、フォーマット名を格納することができる。本実施例では、変換元フォーマット２０１、変換先フォーマット２０２、フォーマット２２２に拡張子を格納するものとする。

【００４５】トークン表１２３は、トークンを管理する表である。１つのトークンは、ある変換元ファイルと、当該変換元ファイルから変換可能な変換先ファイル群からなる一群のファイルに対して割り当てられる。図２に示すように、トークン表１２３は１つ以上のトークン表エントリ２３０からなり、１つのトークン表エントリ２３０が１つのトークンに対応する。トークン表エントリ２３０はさらにトークンＩＤ２３１とファイルＩＤ２３２からなる。トークンＩＤ２３１はトークンを一意に識別する番号、ファイルＩＤ２３２は、現在トークンを保持しているファイルのファイルＩＤである。モード２３３は、当該ファイルの現在のオープン・モードを表わす。

【００４６】削除候補表１２４は、変換先ファイルを列挙した表である。二次記憶装置の空き領域を確保するなどの目的で、変換先ファイルを削除する際に用いる。図２に示すように、削除候補表１２４は１つ以上の削除候補表エントリ２４０からなり、１つの削除候補表エントリ２４０が１つの削除可能なファイルに対応する。削除候補表エントリ２４０は削除可能なファイルのＩＤであるファイルＩＤ２４１を持つ。

【００４７】図１のフォーマット変換制御部１１０は、アプリケーション１０１、１０２から発行されるファイル入出力ＡＰＩおよびファイル管理ＡＰＩの呼び出し（ＡＰＩコール）を受け取り、フォーマット変換を制御する部分である（制御の詳細は後述する）。

【００４８】設定プログラム１０４は、ファイルシステム１００によるフォーマット変換の動作をユーザが設定、変更するためのフォーマット変換設定ＡＰＩを、フォーマット変換設定部１１１に対して発行するプログラムである。フォーマット変換設定部１１１は、ユーザからのフォーマット変換設定ＡＰＩを受けて（１６１）、変換表１２０とファイル表１２２を変更する。フォーマット変換設定ＡＰＩは、変換表１２０の各エントリ２００、およびファイル表１２２の各エントリ２１０を参照し、変更する（１６２）。

【００４９】なお、変換表１２０、ファイル表１２２、名前空間表１２１、トークン表１２３および削除候補表１２４はいずれも、主記憶上か二次記憶上のいずれか片方または両方にあって差し支えない。また、変換表１２０、ファイル表１２２、名前空間表１２１、トークン表１２３および削除候補表１２４はいずれも、ファイルシステム１００の外に保存されても差し支えない。例えば、ファイルシステム１００と別のプログラムがあって、変換表１２０、ファイル表１２２、名前空間表１２１、トークン表１２３および削除候補表１２４の一部または全部に対する参照と更新を提供していて、ファイルシステム１００が当該プログラム経由で各表にアクセスしても差し支えない。

【００５０】動作の説明

以下、第１の実施例の動作を説明する。本発明が特徴を発揮する以下の３つの場面を想定し、順に説明する。

（１）第１のアプリケーション１０１が変換元ファイル１３０を作成する。（２）第１のアプリケーション１０１が変換元ファイル１３０に対して読み書きを行う。

（３）二次記憶装置１１の空き領域が不足した際に、変換先ファイル１３１を消去する。なお、ここでは説明を簡単にするため、第１のアプリケーション１０１と第２

のアプリケーション１０２をとしているが、この数および種類は一例にすぎない。アプリケーションの数および種類はは１つでも、３つ以上でも差し支えない。

【００５１】（１）アプリケーションによるファイルの作成

第１のアプリケーション１０１または第２のアプリケーション１０２は、ファイルシステム１００のファイル作成ＡＰＩを生成対象の第１のファイルの第１のファイル名とともに起動することにより、ファイル作成をファイルシステム１００に依頼する（１５０、１５２）。この際、ファイルシステム１００は図３に示す処理を行う。

【００５２】まず、第１のファイル名を名前空間表１２１に登録する（１５５、ステップ３０１）。具体的には、新たな名前空間表エントリ２１０を名前空間表１２１に割り当て、ファイルＩＤ２１２に他のファイルにまだ割り当てられていない第１のファイルＩＤを格納し、ファイル名２１１に第１のファイル名を格納する。

【００５３】次に、第１のファイルをファイル表１２２に登録する（１５６、ステップ３０２）。具体的には、新たなファイル表エントリ２２０をファイル表１２２に割り当て、ファイルＩＤ２２１に第１のファイルＩＤを、ファイル名の拡張子から決定したフォーマット２２２に第１のファイルの第１のフォーマットを、タイムスタンプ２２３に現在時刻を、変換元ファイルＩＤ２２４に「空」を、トークンＩＤ２２５にまだどのトークンにも割り当てられていない第１のトークンＩＤを、ファイル内容２２６のファイル本体（すなわち変換元ファイル１３０）に「空」をそれぞれ格納する（１５９）。なお、ファイル内容２２６が決定しなければファイルのフォーマットがわからない計算機システムの場合には、例えばフォーマット２２２に「空」をいれておき、後で第１のファイルに対する書き込み等によってファイル内容２２６が得られてから、フォーマット２２２を格納する。

【００５４】次に、第１のファイルに対応するトークンをトークン表１２３に登録する（１５７、ステップ３０３）。具体的には、新たなトークン表エントリ２３０をトークン表１２３に割り当て、トークンＩＤ２３１に第１のトークンＩＤを、ファイルＩＤ２３２に「空」を格納する。

【００５５】次に、１つ以上の第１の変換先ファイル名を第１のファイル名と変換表１２０から生成する（１５４、ステップ３０４）。具体的には、変換表１２０に格納されている変換表エントリ２００のうち、変換元フォーマット２０１が第１のフォーマットに等しい１つ以上の第１の変換表エントリを検索する。そして、第１の変換表エントリのそれぞれについて、第１のファイル名から拡張子を取り去って、代りの拡張子として変換先フォーマット２０２をつけたものを変換先ファイル名とする。これが本実施例におけるファイル名変換方法である。

【００５６】次に、生成した上記変換先ファイル名のそれぞれについて（ステップ３０５の判定がＮの間）、ステップ３０６とステップ３０７の処理を行う。具体的には、ステップ３０１と同様の方法で上記変換先ファイル名を名前空間表１２１に登録し（１５５、ステップ３０６）、変換先ファイルをファイル表１２２に登録する（１５６、ステップ３０７）。変換先ファイルのファイル表１２２への登録では、新たなファイル表エントリ２２０をファイル表１２２に割り当て、ファイルＩＤ２２１に第１のファイルＩＤを、ファイル名の拡張子から決定したフォーマット２２２に第１のファイルの第１のフォーマットを、タイムスタンプ２２３に「空」を、変換元ファイルＩＤ２２４に第１のファイルＩＤを、トークンＩＤ２２５に第１のトークンＩＤを、ファイル内容２２６のファイル本体（すなわち変換先ファイル群１３１、１３１’、…のいずれか）に「空」を格納する（１６０）。また、変換先ファイルがさらに変換元ファイルとなりうる場合、すなわち多段階の変換が可能な場合、変換先ファイルのファイル名をさらにファイル名変換方法で変換し、得られたファイル名およびファイルを本段落の手順で名前空間表１２１とファイル表１２２にそれぞれ登録する。

【００５７】ステップ３０４で生成したすべての変換先ファイル名を処理したら（ステップ３０５の判定がＹ）、残りのファイル生成操作を行う（ステップ３０８）。この操作の中で、例えば二次記憶装置１１へのディスクブロックの割り当て、ファイルの所有者、アクセス権、生成時刻等の属性の設定などを行う場合があるが、これらは既に公知であり、本発明の特徴とも直接関係ないのでこれ以上説明しない。

【００５８】以上のファイル作成ＡＰＩが終了した時点で、ファイルシステム１００は第１のアプリケーション１０１または第２のアプリケーション１０２に対して、結果を返す（１５１、１５３）。この結果はファイルＩＤ（またはファイルＩＤを間接に指す番号）である。

【００５９】以上が第１の実施例において新たなファイルを生成する際の動作である。上述のように、第１の実施例のファイルシステムは、変換元ファイルのファイル名から変換先ファイルのファイル名を得るファイル名変換方法をファイルシステムが持つ。これにより、ユーザがファイル作成ＡＰＩによって直接作成を依頼した第１のファイルが作成されるに加えて、第１のファイルを変換元ファイルとする１つ以上の変換先ファイルがファイルシステムにより生成される。

【００６０】なおこの実施例では、変換元ファイルの作成操作を契機として変換先ファイルのファイル名を生成したが、本発明はこの契機に限定されるものではない。別の契機として、例えばユーザがディレクトリの表示操作または検索操作を用いることが可能である。変換先フ

ァイルのファイル名が生成されるディレクトリに「ファイル名生成を行う必要あり」の印をつけておき、後に当該ディレクトリを表示または検索する際に、実際のファイル名生成を行う。こうすることで、ファイル名生成の処理を遅延し、変換元ファイルの作成操作をより高速にすることが可能である。また、「ファイル名生成を行う必要あり」の印がついたファイルを、一定時間後、または一定時間毎に探し出して、変換先ファイルのファイル名を生成しても良い。

【００６１】（２）アプリケーションによるファイルへの読み書き

アプリケーション群（１０１、１０２）による第１のファイルへの書き込みは、オープンＡＰＩを用いて書き込みモードで第１のファイルをオープンし、書き込みＡＰＩまたは読み出しＡＰＩを用いてデータを読み書きし、クローズＡＰＩを用いて第１のファイルをクローズする、という手順で行う。なお、計算機システムによってはオープンＡＰＩやクローズＡＰＩがない場合も考えられる。このような場合には、読み出しＡＰＩ、書き込みＡＰＩが毎回、オープンとクローズをそれぞれのＡＰＩ処理の最初と最後に行われるとみなして本発明を適用することができる。

【００６２】オープンＡＰＩの動作を、図４を用いて説明する。第１の実施例のオープンＡＰＩは、第１のファイルの第１のファイル名とオープンモード（読み出し、書き込み、または両方）を伴ってアプリケーション群（１０１、１０２）から呼び出される（１５０、１５２）。

【００６３】まず、第１のファイルと変換可能なファイル群の読み書きの一貫性を保持するために、トークンを取得する（ステップ４０１）。具体的には、名前空間表１２１の中から、ファイル名２１１が第１のファイル名に等しい名前空間表エントリ２１０を検索し（１５４）、当該第１の名前空間表エントリのファイルＩＤ２１２に等しいファイルＩＤ２２１を持つファイル表エントリ２２０を検索し（１５６）、当該第１のファイル表エントリのトークンＩＤ２２５に等しいトークンＩＤ２３１を持つトークン表エントリ２３０をトークン表１２３の中から検索する（１５７）。そして、当該第１のトークン表エントリのファイルＩＤ２３２が「空」になるまで待ち、ファイルＩＤ２３２に第１のファイル表エントリのトークンＩＤ２２５を格納する。モード２３３には、オープンモードを格納する。本実施例では、トークンで変換可能なファイル群へのファイル操作をすべて直列化しているが、２つ以上の書き込みＡＰＩ（または２つ以上の書き込みオープンＡＰＩ）を禁止するトークン取得・解放アルゴリズムを使用しても差し支えない。

【００６４】次に、第１のファイルが変換先ファイルか（Ｙ）否か（Ｎ）を判定する（ステップ４０２）。具体的には、前記第１のファイル表エントリの変換元ファイルＩＤ２２４が「空」でなければ、第１のファイルは変換先ファイルである。

【００６５】第１のファイルが変換先ファイルでなければ（ステップ４０２の判定がＮ）、後述のステップ４０８に制御を移す。一方、第１のファイルが変換先ファイルであれば（ステップ４０２の判定がＹ）、第１のファイルの第１の変換元ファイルを決定する（ステップ４０３）。具体的には、第１のファイル表エントリの変換元ファイルＩＤ２２４が第１の変換元ファイルのファイルＩＤである。当該ファイルＩＤに等しいファイルＩＤ２２１を持つファイル表エントリ２２０を検索し、当該第２のファイル表エントリとする。

【００６６】次に、変換先ファイルである第１のファイルが第１の変換元ファイルの最新の内容を反映しているかを確認する。第１のファイル表エントリのタイムスタンプ２２３が「空」、または、「第１のファイル表エントリのタイムスタンプ２２３＜第１のファイル表エントリのタイムスタンプ２２３」であれば（ステップ４０４の判定がＹ）、最新の内容を反映していない。

【００６７】この場合にはまず、変換元ファイルをオープンし（ステップ４０５）、変換プログラムを選択して実行する（ステップ４０６）。具体的には、変換表１２０の中から、変換先フォーマット２０２が第１のファイル表エントリのフォーマット２２２に等しく、かつ変換元フォーマット２０１が第２のファイル表エントリのフォーマット２２２に等しい変換表エントリ２００を検索し（１５４、１５６）、第２のファイル表エントリのファイルＩＤ２２１で示すファイルをオープンし（ステップ４０５）する。そして、当該第１の変換表エントリの変換プログラム２０３を起動する（１６７、（ステップ４０６）。この際、変換プログラム２０３への入力として第２のファイル表エントリのファイル内容２２６のファイル本体を与え（１６３）、変換プログラム２０３の出力を第１のファイル表エントリのファイル内容２２６のファイル本体に格納する（１６４）。この際、第２のファイル表エントリのファイル内容２２６のファイル本体がファイルシステム１００中になければ、二次記憶装置１１から取出す（１６５）。また変換プログラム２０３終了後、第１のファイル表エントリのファイル内容２２６のファイル本体を、二次記憶装置１１に書き出す（１６６）。なお、第１のファイル表エントリのファイル内容２２６のファイル本体を二次記憶装置１１に書き出すことは、必ずしも必要ではない。変換が終わったら、第２のファイル表エントリのファイルＩＤ２２１で示すファイルをあとで述べる方法でクローズする（ステップ４０７）。

【００６８】そして、変換先ファイルのタイムスタンプを現在時刻に設定する（１５６、ステップ４０８）。具体的には、第１のファイル表エントリのタイムスタンプ２２３に現在時刻を格納する。

【００６９】さらに、ステップ４０２で変換先ファイルのタイムスタンプが「空」であったならば、変換先ファイルを削除候補表１２４に登録する（１５８、ステップ４０９）。具体的には、削除候補表１２４に新たな削除候補表エントリ２４０を割り当て、当該削除候補表エントリのファイルＩＤ２４１に第１のファイル表エントリのファイルＩＤ２２１を格納する。

【００７０】一方ステップ４０４の判定がＮなら、第１のファイルは第１の変換元ファイルの最新の内容を反映しているので、ステップ４１０で残りのファイル・オープン操作を行う。この操作の中で、ファイル・ポインタの初期化、主記憶上の読み書き用バッファの割り当て、ファイルアクセス権限の確認などを行う場合があるが、これらは既に公知であり、本発明の特徴とも直接関係ないのでこれ以上説明しない。

【００７１】以上のオープンＡＰＩが終了した時点で、ファイルシステム１００は当該オープンＡＰＩを呼び出したアプリケーションに対し、当該オープンＡＰＩが完了したことを通知する（１５１、１５３）。

【００７２】次に、書き込みＡＰＩの動作を説明する。書き込みＡＰＩは、従来のファイルシステムと同様、アプリケーションがファイル名またはファイルＩＤ（またはファイルＩＤを間接に指す番号）を指定して、ファイルシステム１００にデータの書き込みを依頼する（１５０、１５２）。この際ファイルシステム１００は、与えられたものがファイル名であれば上述のオープンＡＰＩと同様名前空間表１２１からファイルＩＤを得（１５５）、またが与えられたのがファイルＩＤであればそのファイルＩＤを用いる。そして、当該ファイルＩＤから当該ファイルＩＤに対応するファイル表エントリ２２０を得て、当該ファイル表エントリのファイル内容２２６のファイル本体に、与えられたデータを書き込む（１５６）。以上の書き込みＡＰＩが終了した時点で、ファイルシステム１００は当該書き込みＡＰＩを呼び出したアプリケーションに対し、当該書き込みＡＰＩが完了したことを通知する（１５１、１５３）。

【００７３】次に、読み出しＡＰＩの動作を説明する。読み出しＡＰＩは、従来のファイルシステムと同様、アプリケーションがファイル名またはファイルＩＤ（またはファイルＩＤを間接に指す番号）を指定して、ファイルシステム１００にデータの読み出しを依頼する（１５０、１５２）。この際ファイルシステム１００は、与えられたものがファイル名であれば上述のオープンＡＰＩと同様名前空間表１２１からファイルＩＤを得（１５５）、またが与えられたのがファイルＩＤであればそのファイルＩＤを用いる。そして、当該ファイルＩＤから当該ファイルＩＤに対応するファイル表エントリ２２０を得て、当該ファイル表エントリのファイル内容２２６のファイル本体から、与えられたデータを読み出す（１５６）。以上の読み出しＡＰＩが終了した時点で、ファイルシステム１００は当該読み出しＡＰＩを呼び出したアプリケーションに対し、当該読み出しＡＰＩが完了したことを通知する（１５１、１５３）。

【００７４】次に、クローズＡＰＩの動作を図５を用いて説明する。

【００７５】第１の実施例のクローズＡＰＩは、オープンＡＰＩでオープンされた前記第１のファイルの第１のファイルＩＤを伴ってアプリケーション群（１０１、１０２）から呼び出される（１５０、１５２）。

【００７６】第１のファイルのオープンが書き込みモードであって、第１のファイルが変換元ファイルであるかを検査する（ステップ５０１）。オープンが書き込みモードかどうかは、以下の方法で検査する。第１のファイルＩＤに等しいファイルＩＤ２２１を持つファイル表エントリ２２０を検索し（１５６）、当該第１のファイル表エントリのトークンＩＤ２２５に等しいトークンＩＤ２３１を持つトークン表エントリ２３０をトークン表１２３の中から検索する（１５７）。そして、当該第１のトークン表エントリのモード２３３が書き込みモードかを検査する。また、第１のファイル表エントリの変換元ファイルＩＤ２２４が「空」ならば、第１のファイルが変換元ファイルであり、「空」以外ならば第１のファイルは変換先ファイルである（１５６）。

【００７７】ステップ５０１がＹなら、ステップ５０２以下を実行する。具体的には、第１のファイル表エントリのタイムスタンプ２２３に現在時刻を格納する（１５６）。一方ステップ５０１がＮなら、ステップ５０３に制御を移す。

【００７８】次に、トークンを解放する（ステップ５０３）。具体的には、第１のトークン表エントリのファイルＩＤ２３２とモード２３３に「空」を格納する（１５７）。

【００７９】最後に、ステップ５０４で残りのファイル・クローズ操作を行う。この操作の中で、主記憶上の読み書き用バッファの解放、ファイル内容の二次記憶装置１１への書き出しなどを行う場合があるが、これらは既に公知であり、本発明の特徴とも直接関係ないのでこれ以上説明しない。

【００８０】以上のクローズＡＰＩが終了した時点で、ファイルシステム１００は当該クローズＡＰＩを呼び出したアプリケーションに対し、当該クローズＡＰＩが完了したことを通知する（１５１、１５３）。

【００８１】以上のように第１の実施例のファイルシステムでは、主に変換表１２０に変換元ファイルと変換先ファイルと変換プログラムとの間の対応関係（１段階でも、多段階でも良い）をファイルシステム中に保持して、どのファイルをどのファイルへ変換するかをユーザがその都度指示する必要をなくしている。

【００８２】また、第１の実施例はファイルへの読み書きに際して必ずアプリケーションが行う動作（オープン

ＡＰＩとクローズＡＰＩ、読み出しＡＰＩや書き込みＡＰＩでも差し支えない）を、フォーマット変換の契機としても用いる。これにより、第１の実施例のフォーマット変換方法は、アプリケーションの動作をファイルシステムを通じて逐一知り、対応する動作を行うことができる。翻ってアプリケーションは、フォーマット変換方法の動作をまったく意識することがないが、ユーザの作業を進めるために必要な変換は順次行われる。

【００８３】加えて第１の実施例では、主にトークン表１２３とタイムスタンプ２２３を用いて、変換元ファイルの最新の内容を変換先ファイルに反映させ、かつ、アプリケーションが変換元ファイルと変換先ファイルに対して同時に読み書きを要求した場合でも、これらを調停して変換元ファイルと変換先ファイルの一貫性を保持する。

【００８４】（３）変換先ファイルの消去

二次記憶装置１１の空き領域が切迫した場合、変換先ファイル群１３１、１３１’、…を消去することができる。これは以下の手順で行う。

【００８５】ファイルシステム１００は定期的に二次記憶装置１１の空き領域の監視を行い、空き領域の量が一定以下になったら、削除候補表１２４に含まれる削除候補表エントリ２４０のそれぞれについて（１５８）、以下の処理を行う。

【００８６】削除候補エントリのファイルＩＤ２４１と等しいファイルＩＤ２２１を持つファイル表エントリ２２０をファイル表１２２から検索する（１５６）。そして、当該ファイル表エントリのトークンＩＤ２２５と等しいトークンＩＤ２３１を持つトークン表エントリ２３０をトークン表１２３から検索し（１５７）、当該トークン表エントリのファイルＩＤ２３２が当該ファイル表エントリのファイルＩＤ２２１と異なっていれば、当該ファイル表エントリのファイル内容２２６のファイル本体を「空」にし、タイムスタンプ２２３に「空」を格納する（１５６）。

【００８７】以上の操作によって、二次記憶装置１１の空き領域が切迫した場合に、現在使用されていない変換先ファイルのファイル内容をすべて削除し、空き領域を増やすことができる。変換先ファイルのファイル内容を削除したとしても、新たにユーザが変換先ファイルを参照した場合には、上述のようにフォーマット変換がユーザの関与なしに起動され、当該変換先ファイルのファイル内容は再び充填される。このため、ユーザは変換先ファイルのファイル内容が削除されていたことをまったく意識せずに作業を進めることができる。

【００８８】フォーマット変換のタイミング

本実施例では、オープンＡＰＩとクローズＡＰＩの処理中にフォーマット変換を起動しているが、ＡＰＩを契機として、他の起動タイミングでフォーマット変換を行っても差し支えない。

【００８９】例えば、変換元ファイルの書き込みＡＰＩのあと、一定時間経過後にフォーマット変換を行ってもよい。例えばアプリケーションが書き込みＡＰＩを行った時点で、ファイルシステム１００がタイマーをＴ１秒後（Ｔ１は所定の秒数）に設定して、アプリケーションに制御を移す。そして、Ｔ１秒以内に続く書き込みＡＰＩが発行されれば、再びタイマーをＴ１秒後に設定し直す。一方、Ｔ１秒以内に続く書き込みＡＰＩが発行されなければ、フォーマット変換を行う。このようにすると、書き込みＡＰＩの発行が連続して行われる間はフォーマット変換を控え、書き込みＡＰＩの発行が止んでしばらくしたところで、フォーマット変換を行って、変換先ファイル群に最新の内容を伝播する、という動作ができる。

【００９０】また、第１の実施例ではオープンＡＰＩの発行に同期してフォーマット変換を行っていたが、上記の書き込みＡＰＩと同様に、クローズＡＰＩの処理後しばらくしてから、ファイルシステム１００が自発的にフォーマット変換を行うことも可能である。こうすれば、オープンＡＰＩの実行時にはフォーマット変換を行わなくてもよい場合が増えるので、オープンＡＰＩがより高速に行えるようになる。自発的なフォーマット変換のタイミングとしては、クローズＡＰＩのＴ２秒後（Ｔ２は所定の秒数）、通常業務が終了する深夜Ｓ時から（Ｓは適当な時刻）、計算機１０のＣＰＵ負荷がＬ以下になった時点（Ｌは適当なＣＰＵ負荷指標）で、などの選択肢が考えられ、どれも有効である。これらの自発的なフォーマット変換は、ＡＰＩ発行を契機としたフォーマット変換と併用することで、変換先ファイルが最新であることを保証するのが望ましい。

【００９１】第２の実施例：２段階変換

第１の実施例と異なる本発明の形態を、図面を参照しながら説明する。以下に述べる発明の実施の形態を「第２の実施例」と呼ぶことにする。

【００９２】全体構成

図６を用いて第２の実施例の構成を説明する。第２の実施例は、第１の実施例の主要部分を、分散システム（２つ以上の計算機がネットワークで結合された計算機システム）に拡張したものである。

【００９３】なお、図６に示した計算機、ネットワーク、アプリケーション、変換プログラム、変換元ファイル、変換先ファイルの数と構成は、一例に過ぎず本発明はこれに限定されるものではない。また、図６はネットワーク１２を介して２つの計算機１０、１０’が結合した構成をとっているが、第２の実施例で行っている処理を、ひとつの計算機上で実現しても差し支えない。

【００９４】計算機１０’は第１の実施例で説明した計算機１０と同様、いわゆるパーソナル・コンピュータ、ワークステーション、並列計算機、大型計算機等、任意のコンピュータでよい。図６には二次記憶装置を図示し

ていないが、計算機１０も計算機１０’も二次記憶装置を持っていてよい。

【００９５】ネットワーク１２は、計算機１０と計算機１０’、または他の計算機を結合する。ここで、ネットワーク１２はある団体（企業や学校や類似の団体）の全体や一部門でよく使用されるＬＡＮやＷＡＮ（イントラネットと呼ばれる場合がある）でもよく、また地理的に分散した複数の地点を結合するＷＡＮの一部または全部でもよい。すなわち、ネットワーク１２はインターネットの一部または全部でよい。インターネットは、アメリカ合衆国を中心に発展したコンピュータ・ネットワークである。通信プロトコルとして主にＴＣＰ／ＩＰ（Ｔｒａｎｓｍｉｓｓｉｏｎ Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ／Ｉｎｔｅｒｎｅｔ Ｐｒｏｔｏｃｏｌの略）を用いる。またネットワーク１２は、計算機間結合網や並列計算機内部のプロセッサ要素間の結合網でもよい。

【００９６】ファイルシステム１００’は第１の実施例で説明したファイルシステム１００と同じ機能を持つソフトウェアである。フォーマット変換制御部１１０’はフォーマット変換制御部１１０と、それぞれ同じ機能を持つソフトウェアであるが、以下に述べるようにクローズＡＰＩの動作が第１の実施例とは異なる。

【００９７】ファイルシステム１００とファイルシステム１００’はそれぞれ変換表１２０、ファイル表１２２、名前空間表１２１、トークン表１２３、削除候補表１２４を持つ（図６には示していない）。

【００９８】動作の説明

以下、第２の実施例の動作を説明する。第２の実施例の特徴として、アプリケーション１０１が変換元ファイル１３０を作成して書き込み、アプリケーション１０２が変換元ファイル１３０から変換された変換先ファイル１３１を読み出す場合を想定して説明を行う。

【００９９】アプリケーション１０１は、ファイルシステム１００のファイル作成ＡＰＩを生成対象の第１のファイルの第１のファイル名とともに起動することにより、変換元ファイル１３０の作成をファイルシステム１００に依頼する（１５０）。この際、ファイルシステム１００は第１の実施例ですでに説明した処理を図３のフローチャートに従って行う。ここで変換元ファイル１３０は、図６に示すように変換プログラム６０３および中間ファイル１３０’に対応づけられている。フォーマット変換制御部１１０はオープンＡＰＩの途中で中間ファイルとして用いる中間ファイル１３０’を生成するためにフォーマット変換制御部１１０’と通信を行い（６５０）、フォーマット変換制御部１１０’が中間ファイル１３０’を生成する。

【０１００】次に、アプリケーション１０１がファイルシステム１００に対して、変換元ファイル１３０をオープンし、書き込みを行い、クローズする（１５０）。この一連の操作で変換元ファイル１３０のファイル内容が格納される。変換元ファイル１３０のオープンＡＰＩは図４を用いてすでに説明した手順で行う。また、変換元ファイル１３０の書き込みＡＰＩも、第１の実施例ですでに説明した手順で行う。続くアプリケーション１０１へのクローズＡＰＩの処理は、フォーマット変換制御部１１０が図７にしたがって行う。以下、この手順を説明する。

【０１０１】第２の実施例の変換元ファイル１３０のクローズＡＰＩは、オープンＡＰＩでオープンされた変換元ファイル１３０の第１のファイルＩＤを伴ってアプリケーション群（１０１、１０２）から呼び出される（１５０）。クローズの対象となっている第１のファイルのオープンが書き込みモードであって、第１のファイルが変換元ファイルであるか（Ｙ）否か（Ｎ）を検査する（ステップ７０１）。オープンが書き込みモードかどうかは以下の方法で検査する。第１のファイルＩＤに等しいファイルＩＤ２２１を持つファイル表エントリ２２０を検索し、当該第１のファイル表エントリのトークンＩＤ２２５に等しいトークンＩＤ２３１を持つトークン表エントリ２３０をトークン表１２３の中から検索する。そして、当該第１のトークン表エントリのモード２３３が書き込みモードかを検査する。また、第１のファイル表エントリの変換元ファイルＩＤ２２４が「空」ならば、第１のファイルが変換元ファイルであり、「空」以外ならば第１のファイルは変換先ファイルである。ステップ７０１の判定がＹなら、ステップ７０２から７０７を実行する。一方、ステップ７０１の判定がＮなら、ステップ７０３とステップ７０４を実行する。

【０１０２】ステップ７０２では、第１のファイル表エントリのタイムスタンプ２２３に現在時刻を格納する（１５６）。

【０１０３】次に、トークンを解放する（ステップ７０３）。具体的には、第１のトークン表エントリのファイルＩＤ２３２とモード２３３に「空」を格納する（１５７）。

【０１０４】次に、ステップ７０４で残りのファイル・クローズ操作を行う。この操作の中で、主記憶上の読み書き用バッファの解放、ファイル内容の二次記憶装置１１への書き出しなどを行う場合があるが、これらは既に公知であり、本発明の特徴とも直接関係ないのでこれ以上説明しない。

【０１０５】次に、ステップ７０５で、第１のファイルに対する変換先ファイルを決定する。具体的には、ファイル表１２２の中で変換元ファイルＩＤ２２４が第１のファイルＩＤに等しいファイル表エントリ２２０を検索する。当該ファイル表エントリが変換先ファイルのファイル表エントリである。

【０１０６】次に、ステップ７０６で、当該変換先ファイルを読み出しモードでオープンする。このステップの処理は図４ですでに説明したものである。

【０１０７】次に、ステップ７０７で、第１のファイルをクローズする。このステップの処理は図５ですでに説明したものである。

【０１０８】以上のクローズＡＰＩが終了した時点で、ファイルシステム１００は当該クローズＡＰＩを呼び出したアプリケーションに対し、当該クローズＡＰＩが完了したことを通知する（１５１）。特にステップ７０６で、変換先ファイルをオープンしている点が、このオープンＡＰＩ処理の特徴である。このオープンによって、変換プログラム６０３によって変換元ファイル１３０の最新情報が中間ファイル１３０’に伝播する。加えてこのオープンによって、中間ファイル１３０’を変換元ファイルとした変換先ファイルのファイル名が名前空間表１２１に登録される。

【０１０９】次に、アプリケーション１０２がファイルシステム１００’に対して、変換先ファイル１３１をオープンし、読み出しを依頼する（１５２）。この時、ファイルシステム１００’は図４にしたがって、第１の実施例ですでに説明した処理を行う。ただ第２の実施例では、変換先ファイル１３１は変換プログラム１０３と中間ファイル１３０’に対応づけられている。このため、変換先ファイル１３１のオープンＡＰＩ操作は、中間ファイル１３０’を入力変換先ファイル１３１を出力として変換プログラム１０３を起動する、というフォーマット変換操作を引き起こす。

【０１１０】以上が、アプリケーション１０１が変換元ファイル１３０を作成して書き込み、アプリケーション１０２が変換元ファイル１３０を作成して書き込み、アプリケーション１０２が変換元ファイル１３０から変換された変換先ファイル１３１を読み出す場合の第２の実施例の動作である。

【０１１１】第２の実施例では、計算機１０にある変換元ファイル１３０への変更が、すぐに計算機１０’に伝播する。この結果、計算機１０の電源がオフになったり、計算機１０の二次記憶装置が故障したりして、変換元ファイル１３０が一時的または永続的に利用不能になっても、計算機１０’の中間ファイル１３０’、変換先ファイル１３１、１３１’、…は計算機１０’上のアプリケーションから利用可能である。

【０１１２】この特徴は、特に計算機１０がユーザのよく利用するパーソナル・コンピュータや携帯型コンピュータで、計算機１０’が多数のユーザが利用するWWWサーバなどの共有情報サーバである場合に特に効果がある。なぜなら、計算機１０のユーザが計算機１０の電源を切断したり、ネットワークから計算機１０を切断しても、計算機１０’経由で変換元ファイル１３０の情報を共有しているユーザにはなんら問題ないし、また、計算機１０’経由で変換元ファイル１３０の情報を共有しているユーザがいる間でも、これらのユーザに変換先ファイル１３１、１３１’、…の利用を中断させることなく、計算機１０のユーザが変換元ファイル１３０を更新することができるためである。この場合、変換プログラム６０３は、単なるコピープログラムでも差し支えない。

【０１１３】変形例１：ファイル単位の変換指定

上記第１、第２の実施例では変換表１２０を用いて、変換元フォーマットから変換先フォーマットへの対応を行っている。しかし、本発明の実施の形態はフォーマット単位の対応づけに限定されるものではない。特に変形例１として、変換表に変換元ファイルから変換先ファイルへの対応づけを格納することができるようにしても差し支えない。これを行うには、変換表エントリ２００の内部構造を、変換元フォーマット２０１と変換先フォーマット２０２と変換プログラム２０３の組ではなく、変換元ファイルＩＤと変換先ファイルＩＤと変換プログラム２０３の組とし、オープンＡＰＩやクローズＡＰＩで変換表１２０を検索する時に、ファイルＩＤを用いればよい。さらに進んで、実施例１の変換表１２０と変形例１の変換表の２つの形式を、１つの変換表に混在させる実施の方法でも差し支えない。

【０１１４】変形例２：変換元ファイルの隠蔽

ユーザの都合によっては、ファイルシステムが変換元ファイルを隠し、変換先ファイルのみを提供したほうが良い場面もある。例えば、第１の実施例で、変換元ファイル１３０が読み書き可能なフォーマットで、変換先ファイル群１３１、１３１’、…が読み出しのみ可能なフォーマットである場合、変換元ファイル１３０を作成したユーザは他のユーザに変換先ファイル群１３１、１３１’、…のみアクセスさせることで、変換元ファイル１３０を変更から守ることができる。

【０１１５】また第２の実施例では、計算機１０’のユーザは中間ファイル１３０’と変換先ファイル１３１、１３１’、…とのいずれにもアクセスが可能である。しかし中間ファイル１３０’はファイルシステム１００が使用する中間ファイルなので、中間ファイル１３０’を隠した方がユーザにとってわかりやすいと思われる。

【０１１６】このように、変換元ファイル（または特定の変換先ファイル）をユーザから隠蔽する機能を、上記第１の実施例、第２の実施例、および変形例１に追加することができる。具体的には、ファイル表１２２に新たな項目として「隠蔽変換元フラグ」と「隠蔽変換先リスト」を追加する。前者は変換元ファイルを隠蔽するため、また後者は変換先ファイルを隠蔽するためである。あるファイルの隠蔽変換元フラグが真なら、ファイルシステムは名前空間表１２１から当該ファイルのファイル名を削除する。また、隠蔽変換先リストは変換先フォーマットのリストを格納しており、上記ファイル作成ＡＰＩにおいて、変換先フォーマットが、変換元ファイルの隠蔽変換先リストに含まれていれば、ステップ３０６において当該変換先フォーマットのファイルを名前空間表

１２１に追加するのを止める。

【０１１７】パーソナル・コンピュータ（ＰＣ）への適用例

本発明をパーソナル・コンピュータ（ＰＣ）へ適用した例を、図８を用いて説明する。なお、計算機やアプリケーションの種類や数は、例として用いたものであり、本発明の範囲を限定するものではない。

【０１１８】ＰＣ　８００はユーザが利用する計算機である。例えばワード・プロセッサ８０１やWWWブラウザ８０２やｐｄｆ文書表示プログラム８０３などが、ＰＣ　８００上のアプリケーションとして動作する。本発明に基づくファイルシステム１００は、上述の説明通り、アプリケーションのファイル操作に伴って、必要なフォーマット変換を行う。例えば、ユーザがワード・プロセッサ８０１で作成した文書を保存する場合、ｈｅｌｌｏ.ｄｏｃ　８０４なるファイルを作成し、書き込みを行って二次記憶装置１１に文書を保存する（８１１）。この際、ファイルシステム１００は、図３のフローチャートに従って、変換先ファイルのファイル名を図１の名前空間表１２１（図８には図示していない）に登録する。この結果、ユーザは変換先ファイル（図８の例ではｈｅｌｌｏ.ｈｔｍｌ　８０５とｈｅｌｌｏ.ｐｄｆ　８０６）を操作できるようになる。例えば、ユーザがWWWブラウザ８０２を用いてｈｅｌｌｏ.ｈｔｍｌ　８０５を参照すると、WWWブラウザ８０２はｈｅｌｌｏ.ｈｔｍｌ　８０５を読み出す準備として、オープンＡＰＩを発行する（８２１）。この際、ファイルシステム１００は図４のフローチャートに従って、変換プログラム８２３を決定し、変換元ファイルから変換先ファイルへの変換（８２２）を行って変換元ファイルであるｈｅｌｌｏ.ｄｏｃ８０４から変換先ファイルであるｈｅｌｌｏ.ｈｔｍｌ　８０５の内容を得る。これによってWWWブラウザ８０２は、続くｈｅｌｌｏ.ｈｔｍｌ　８０５への読み出しＡＰＩ（８２４）によって、先ほどワード・プロセッサ８０１が作成したｈｅｌｌｏ.ｄｏｃ　８０４の内容を、異なるフォーマットを持つファイルｈｅｌｌｏ.ｈｔｍｌ　８０５として得ることができる。なお、この一連の動作に、ユーザやアプリケーションはフォーマット変換にかかわる操作を行う必要が一切なかった点に留意すべきである。

【０１１９】本発明はアプリケーションによらないので、別のアプリケーション（例えばｐｄｆ文書表示プログラム８０３）が別の変換先ファイルの読み出しを行っても、まったく問題ない。この場合、先ほどと同様、オープンＡＰＩ（８２５）の発行に同期して変換プログラム８２７を決定し、変換元ファイルであるｈｅｌｌｏ.ｄｏｃ　８０４から変換先ファイルであるｈｅｌｌｏ.ｐｄｆ　８０６へのフォーマット変換（８２６）を行う。これによってｐｄｆ文書表示プログラム８０３は、続くｈｅｌｌｏ.ｐｄｆ　８０６への読み出しＡＰＩ（８２８）を行うことができる。

【０１２０】なお、現在ＰＣで使用されているＯＳのなかには、従来のＯＳの機能を、複数のダイナミック・リンク・ライブラリ（ＤＬＬ）の集合として実現しているＯＳがある。このようなＯＳ上ではファイルシステムの主要部分もＤＬＬとなっている。本発明をこのようなＤＬＬベースのＯＳに適用する１つの方法は、ファイルシステム機能を提供するＤＬＬ（仮にＦＳ.ＤＬＬと呼ぶ）を本発明に基づくファイルシステムの機能を提供するＤＬＬ（仮にＮＥＷ・ＦＳ.ＤＬＬと呼ぶ）で置き換える方法である。ＮＥＷ・ＦＳ.ＤＬＬの中で従来のファイルシステムと同じ動作を行う部分については、ＦＳ.ＤＬＬの機能を関数呼び出しにより呼ぶことにすれば、既にある機能を再び実現することなく本発明の機能を追加することができる。すなわち、ＮＥＷ・ＦＳ.ＤＬＬをＦＳ.ＤＬＬの上にかぶせることによって、これまでＦＳ.ＤＬＬが受けていたアプリケーションからのＡＰＩ呼び出しを、ＮＥＷ・ＦＳ.ＤＬＬが受け、ＦＳ.ＤＬＬの機能が必要な場面ではＮＥＷ・ＦＳ.ＤＬＬがＦＳ.ＤＬＬのＡＰＩを起動する。例えば、ステップ４１０、ステップ５０４、ステップ５０４、ステップ５０４、ステップ７０４は、従来のファイルシステムの機能を呼び出す部分なので、これらのステップでＮＥＷ・ＦＳ.ＤＬＬがＦＳ.ＤＬＬを呼び出せば良い。

【０１２１】WWWシステムへの適用例（１）：次に本発明をWWWシステムに適用した例を図９を用いて説明する。ネットワーク１２は、既に説明したように、イントラネットやインターネットを含む計算機間の結合である。本適用例は特にＴＣＰ／ＩＰとＨＴＴＰ（Ｈｙｐｅｒ―Ｔｅｘｔ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌの略）を用いるWWWに良く適合する。

【０１２２】この例では、本発明を組み込んだファイルシステム１００を持つサーバ計算機９００が、WWWサーバ９１１や分散ファイル・サーバ９１２を用いて、ネットワーク１２で結合された機能限定ＰＣ　９０１およびＰＣ９０２にファイルを提供する。機能限定ＰＣ　９０１は、ネットワーク・コンピュータやＴｈｉｎ　ＰＣと呼ばれている機能限定、低価格なパーソナル・コンピュータである。このような計算機は、低価格化を進めるために必要最低限のソフトウェアしか搭載しないのが普通である。ＰＣ　９０２は通常のパーソナル・コンピュータである。なお、この例では分散ファイル・サーバ９１２をファイルシステム１００と異なるプログラムとしているが、これらを１つにした、いわゆる分散ファイル・システムを用いても差し支えない。分散ファイル・システムは分散ファイル・サーバ９１２がファイルシステム１００に組み込まれたものと考えることができる。

【０１２３】ユーザは機能限定ＰＣ　９０１やＰＣ　９０２を操作して、サーバ計算機９００上のファイルにアクセスする。この例では、機能限定ＰＣ　９０１のユー

ザはWWWブラウザ９１６を用い、またＰＣ　９０２のユーザはｐｄｆ文書表示プログラム９１７を用いてアクセスする。また、サーバ計算機９００上で動作するワード・プロセッサ９１０を用いると、サーバ計算機９００上に新たなファイルを作成、または既存のファイルを変更することができる。

【０１２４】ユーザがワード・プロセッサ９１０で作成した文書を保存する場合、ワード・プロセッサ９１０はファイルシステム１００に変換元ファイルであるｈｅｌｌｏ.ｄｏｃ　９１３の作成を依頼し（９２０）、書き込みを行って二次記憶装置１１に文書を保存する（９２１）。この際、ファイルシステム１００は図３のフローチャートに従って、変換先ファイルのファイル名を図１の名前空間表１２１（図９には図示していない）に登録する。この結果、サーバ計算機９００上のアプリケーションである共有情報提供ソフトウェア（この例ではWWWサーバ９１１や分散ファイル・サーバ９１２）は変換先ファイル（図９の例ではｈｅｌｌｏ.ｇｉｆ　９１４とｈｅｌｌｏ.ｐｄｆ　９１５）を操作できるようになる。

【０１２５】例えば、機能限定ＰＣ　９０１のユーザがWWWブラウザ９１６を用いて画像ファイルｈｅｌｌｏ.ｇｉｆ　９１４を参照すると、WWWブラウザ９１６はｈｅｌｌｏ.ｇｉｆ　９１４を読み出す要求をWWWサーバ９１１に送る（９３０）。この要求を受けたWWWサーバ９１１は、ｈｅｌｌｏ.ｇｉｆ　９１４を読み出す準備としてオープンＡＰＩを発行する（９３１）。この際、ファイルシステム１００は図４のフローチャートに従って変換プログラム９３３を決定し、変換元ファイルから変換先ファイルへの変換（９３２）を行ってｈｅｌｌｏ.ｇｉｆ９１４の内容をｈｅｌｌｏ.ｄｏｃ　９１３から得る。これによってWWWブラウザ９１６は、続くｈｅｌｌｏ.ｇｉｆ　９１４への読み出しＡＰＩ（９３４）によって、先ほどワード・プロセッサ９１０が作成したｈｅｌｌｏ.ｄｏｃ９１３の内容を、異なるフォーマットを持つファイルｈｅｌｌｏ.ｇｉｆ　９１４として得ることができる。この結果は、機能限定ＰＣ　９０１上のWWWブラウザ９１６に返される（９３５）。WWWブラウザ９１６は画像ファイルを再生する能力を備えているので、ｈｅｌｌｏ.ｄｏｃ　９１３の内容を機能限定ＰＣ９０１上に表示することができる。すなわち本発明によって、ユーザはｈｅｌｌｏ.ｄｏｃ　９１３を作成したワード・プロセッサを搭載しない機能限定ＰＣ９０１上からでも、ｈｅｌｌｏ.ｄｏｃ　９１３の内容を参照することができる。

【０１２６】同様に、別のユーザがＰＣ　９０２からｐｄｆ文書表示プログラム９１７を用いて変換先ファイルの読み出しを行っても、まったく問題ない。たとえばｐｄｆ文書表示プログラム９１７が分散ファイル・サーバ９１２経由でｈｅｌｌｏ.ｐｄｆ　９１５を読み出す場合、まずｐｄｆ文書表示プログラム９１７が分散ファイル・サーバ９１２にｈｅｌｌｏ.ｐｄｆ　９１５を読み出す要求を送る（９４０）。これを受けた分散ファイル・サーバ９１２は、ｈｅｌｌｏ.ｐｄｆ　９１５をオープンし、読み出そうとする。この際、ファイルシステム１００は上記の説明同様、オープンＡＰＩ（９４１）の発行に同期して変換プログラム９４３を決定し、ｈｅｌｌｏ.ｄｏｃ　９１３からｈｅｌｌｏ.ｐｄｆ　９１５へのフォーマット変換（９４２）を行う。これによって分散ファイル・サーバ９１２は、続くｈｅｌｌｏ.ｐｄｆ　９１５への読み出しＡＰＩ（９４４）を行い、結果をｐｄｆ文書表示プログラム９１７に返すことができる（９４５）。

【０１２７】この例によって、本発明が機能限定ＰＣを含む分散計算機システム上でも有効であることを示した。

【０１２８】WWWシステムへの適用例（２）：さらに、WWWとＰＣ、ＴｈｉｎＰＣを用いた分散計算機システムの別の例を、図１０を用いて説明する。なお、計算機やアプリケーションの種類や数は、一例に過ぎず本発明はこれに限定されるものではない。

【０１２９】ＰＣ　１００１を用いているユーザは、ワード・プロセッサ１０１０を用いて文書を作成する。この例ではｈｅｌｌｏ.ｄｏｃ　１０１３を生成して（１０２０）書き込みを行い（１０２１）、書き込みが完了した時点でクローズを行う。

【０１３０】ｈｅｌｌｏ.ｄｏｃ　１０１３を生成する際、ファイルシステム１００はｈｅｌｌｏ.ｄｏｃ　１０１３の変換先ファイルである中間ファイルｈｅｌｌｏ.ｄｏｃ’１０１４も作成する。さらに図３のフローチャートに従って再帰的にｈｅｌｌｏ.ｐｄｆ　１０１５およびｈｅｌｌｏ.ｇｉｆ　１０１６が作成される。そして最後のクローズの際に、ファイルシステム１００は図７のフローチャートに従って、変換先ファイルである中間ファイルｈｅｌｌｏ.ｄｏｃ’１０１４をオープンし、クローズする。この結果、変換プログラム１０２３によるｈｅｌｌｏ.ｄｏｃ　１０１３からｈｅｌｌｏ.ｄｏｃ’１０１４への変換が行われる（１０２２）。

【０１３１】次に機能限定ＰＣ　１００３のユーザがWWWブラウザ１０１２を用いてｈｅｌｌｏ.ｇｉｆ　１０１６をアクセスする要求をWWWサーバ１０１１に発行すると（１０３０）、この要求を受けたWWWサーバ１０１１はｈｅｌｌｏ.ｇｉｆ１０１６読み出しのためのオープンＡＰＩを発行する（１０３１）。このオープンＡＰＩの処理中にファイルシステム１００’は、変換プログラム１０３３を用いたｈｅｌｌｏ.ｄｏｃ’１０１４からｈｅｌｌｏ.ｇｉｆ　１０１６への変換（１０３２）を行う。この結果、WWWサーバ１０１１が続いて発行するｈｅｌｌｏ.ｇｉｆ　１０１６の読み出しＡＰＩは、ｈｅｌｌｏ.ｄｏｃ’１０１４すなわちｈｅｌ

ｌｏ.ｄｏｃ　１０１３から得られた最新の情報がWWWサーバ１０１１を経由して（１０３４）ユーザに渡される（１０３５）。

【０１３２】サーバ計算機１００２は多くの場合多数のユーザによって共用されているので、２４時間運転で運用される。ＰＣ　１００１は一人のユーザの使用する計算機なので、頻繁に電源がオン、オフされるのが普通である。ＰＣ　１００１の電源がオフになるなどの理由でｈｅｌｌｏ.ｄｏｃ　１０１３にアクセスができない場合でも、本発明が中間ファイルｈｅｌｌｏ.ｄｏｃ’１０１４を介したフォーマット変換機能を備えている結果、機能限定ＰＣ　１００３のユーザは、ｈｅｌｌｏ.ｇｉｆ　１０１６やｈｅｌｌｏ.ｐｄｆ　１０１５、およびｈｅｌｌｏ.ｄｏｃ’１０１４にアクセスすることが可能である。

【０１３３】分散検索システムへの適用例：企業内情報システムにおいて、種々の情報を検索する分散検索システムに本発明を適用した例を図１１を用いて説明する。

【０１３４】企業内外の各所で情報提供を行うWWWサーバ計算機１１０１、１１０２、１１０３、…から、ユーザが所望する特定分野の情報を検索する。検索を行うのは、検索サーバ計算機１１００上の検索プログラムであり、この例では全文検索サーバ１１１１と画像検索サーバ１１１２である。全文検索サーバ１１１１は文字列検索を行い、画像検索サーバ１１１２は画像のパターンマッチによる検索を行う。音声パターン検索、データベース検索など、他の検索方法も混在して構わない。WWWサーバ計算機１１０１、１１０２、１１０３、…および検索サーバ計算機１１００はネットワーク１２で結合されている。ネットワーク１２は企業内のネットワーク（イントラネット）でも、企業間を繋ぐネットワークでも、さらには全世界を繋ぐインターネットのようなネットワークでも差し支えない。

【０１３５】企業内外のWWWサーバ計算機群からは、種々のフォーマットの情報が提供されるのが普通である。そこで検索サーバ計算機１１００に本発明に基づくファイルシステム１００を組み込み、各種フォーマットの違いをアプリケーションに負担をかけずに解消する。図１１では、WWWサーバ計算機１１０１がｈｅｌｌｏ．ｄｏｃ１１２０を、WWWサーバ計算機１１０２がｎｅｗｓ．ｐｄｆ　１１２１を、WWWサーバ計算機１１０３がｓｕｒｖｅｙ.ｇｉｆ　１１２２を、それぞれ提供している。これらはまず検索サーバ計算機１１００上のWWWクライアント１１１０によって収集され（１１２３、１１２４、１１２５）、ファイルシステム１００に格納される（１１２６、１１２７、１１２８）。なお、ファイルシステム１００は二次記憶装置を持って良いが図１１では省略している。

【０１３６】全文検索サーバ１１１１は文字形式ファイル「.ｔｘｔ」を入力として受け取るので、ファイルシステムｉ００は全文検索サーバ１１１１が発行するｈｅｌｌｏ.ｔｘｔ　１１３３、ｎｅｗｓ.ｔｘｔ　１１３６、ｓｕｒｖｅｙ.ｔｘｔ　１１３９へのファイル操作ＡＰＩに応じてｈｅｌｌｏ.ｄｏｃ　１１２９からｈｅｌｌｏ.ｔｘｔ　１１３３を、ｎｅｗｓ.ｐｄｆ　１１３０からｎｅｗｓ.ｔｘｔ１１３６を、ｓｕｒｖｅｙ.ｇｉｆ　１１３１からｓｕｒｖｅｙ.ｔｘｔ　１１３９を、それぞれ対応する変換プログラム（図１１では図示省略）によりフォーマット変換して作成する（１１３２、１１３５、１１３８）。これによって全文検索サーバ１１１１は、本来異なるフォーマットを持つ情報に対しても検索を行うことができるようになり（１１３４、１１３７、１１４０）、かつ検索をかけた情報が別々のフォーマットを持っていたことを気にしなくてよい（すなわち全文検索サーバ１１１１内に特別なプログラミングが必要ない）。さらにユーザは、検索結果をもとに情報が最も豊かな変換元ファイル（ｈｅｌｌｏ.ｄｏｃ　１１２９、ｎｅｗｓ.ｐｄｆ　１１３０、ｓｕｒｖｅｙ.ｇｉｆ　１１３１）の情報にアクセスすることができる。この場合にもファイルシステム１００は、変換元ファイルをユーザが使うアプリケーションが利用可能なフォーマットに変換することにより、ユーザの利便を図ることができる。

【０１３７】さらに、同じ情報源を画像に変換することにより、まったく異なる検索方法を実施することが可能である。画像検索サーバ１１１２は画像形式ファイル「.ｇｉｆ」を入力として受け取るので、ファイルシステム１００は画像検索サーバ１１１２が発行するｈｅｌｌｏ.ｇｉｆ　１１４２、ｎｅｗｓ.ｇｉｆ　１１４５、ｓｕｒｖｅｙ.ｇｉｆ　１１３１へのファイル操作ＡＰＩに応じてｈｅｌｌｏ.ｄｏｃ　１１２９からｈｅｌｌｏ.ｇｉｆ　１１４２を、ｎｅｗｓ.ｐｄｆ　１１３０からｎｅｗｓ.ｇｉｆ　１１４５を、それぞれ対応する変換プログラム（図１１では図示省略）によりフォーマット変換して作成する（１１４１、１１４４）。これによって画像検索サーバ１１１２は、本来異なるフォーマットを持つ変換元ファイル（１１２９、１１３０、１１３１）の情報に対して検索を行うことができ（１１４３、１１４６、１１４７）、かつ検索をかけた情報が別々のフォーマットを持っていたことを気にしなくてよい。

【０１３８】電子商取引システムへの適用例：企業と企業、または企業と個人の間の取り引きを電子的に行う電子商取引（ＥＣ）システムに本発明を適用した例を図１２を用いて説明する。

【０１３９】例えばインターネット経由でＥＣを行う場合には、請求書はいくつもの企業や団体のネットワークを経由して第１のユーザから第２のユーザに届けられる。この際、第１、第２のユーザのプライバシーを保護する目的で、また請求書が第三者によって不当に改竄さ

れるのを防ぐ目的で、請求書を暗号化して送ることがよく行われる。

【０１４０】ＥＣサーバ計算機１２００はＥＣクライアント計算機１２０１と通信して電子商取引を行う。ＥＣサーバ計算機１２００を使用する第１のユーザが、ＥＣクライアント計算機１２０１を使用する第２のユーザの注文に対する請求書を作成し、第２のユーザがそれを受け取る場面を説明する。このような場面での暗号化（および復号化）は、一種のフォーマット変換と考えることができる。この場合の変換プログラムは、暗号化プログラムや復号化プログラムである。

【０１４１】第１のユーザは過去に第２のユーザから受け取った注文に対する請求書であるｉｎｖｏｉｃｅ．ｄｏｃ　１２２０をワードプロセッサ１２１１で作成する（１２３０）。そしてｉｎｖｏｉｃｅ．ｄｏｃ　１２２０を暗号化プログラム１２１２で暗号化し（１２３１）、暗号化後のファイルであるｉｎｖｏｉｃｅ．ｄｏｃ．ｃｒｐ　１２２１を電子メール（または電子メールの添付ファイル）で第２のユーザに送る（１２３３）。これを受け取った第２のユーザは、電子メールをファイルシステムにｉｎｖｏｉｃｅ．ｄｏｃ．ｃｒｐ　１２２１’として保存する。この際、本発明のファイルシステムが、復号化プログラム１２１３を変換プログラムとし、ｉｎｖｏｉｃｅ．ｄｏｃ．ｃｒｐ　１２２１’を変換元ファイルとしたフォーマット変換操作を行い（１２３４）、変換先ファイルとしてｉｎｖｏｉｃｅ．ｄｏｃ　１２２０’を第２のユーザに提供する（１２３５）。これにより第２のユーザは、ｉｎｖｏｉｃｅ．ｄｏｃ　１２２０’を手動で復号化する手間を負うことなく、ワードプロセッサ１２１１’を用いて請求書を参照することができる（１２３６）。またこの例では示さなかったが、第１のユーザが請求書の暗号化を行う場面でも本発明を適用することができる。これにより、第１のユーザ、第２のユーザとも請求書の暗号化と復号化を気にする必要なく、高速かつ快適に電子商取引を行うことができる。

【０１４２】

【発明の効果】ユーザはアプリケーション上で行うユーザの本来の作業を行えばよくなり、その過程で必要となる様々なフォーマット変換（１段階でも、多段階でも良い）はユーザの関与なしで行われる。この際、変換元ファイルを指定したり、フォーマット変換のタイミングを指定する必要がなく、ユーザは常に最新の変換先ファイルを利用可能となる。さらに、変換元ファイルと変換先ファイル群との間の一貫性が保持される。また変換先ファイルを多く保持することによる二次記憶領域の無駄を解消できる。万一変換元ファイルが利用不能な間もフォーマット変換が行える。

【図面の簡単な説明】

【図１】第１の実施例の内部構成の概略を示すブロック図。

【図２】第１の実施例で用いるデータ構造を示す図。

【図３】ファイル作成ＡＰＩの動作を説明するフローチャート。

【図４】オープンＡＰＩの動作を説明するフローチャート。

【図５】クローズＡＰＩの動作を説明するフローチャート。

【図６】第２の実施例の内部構成の概略を示すブロック図。

【図７】第２の実施例で変換元ファイルのクローズＡＰＩの動作を説明するフローチャート。

【図８】本発明のパーソナル・コンピュータへの適用例を示す図。

【図９】本発明のWWWシステムへの適用例を示す図。

【図１０】本発明のWWWシステムへの別の適用例を示す図。

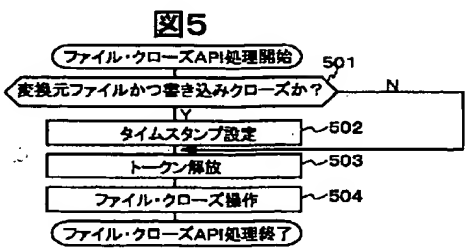
【図１１】本発明の分散検索システムへの別の適用例を示す図。

【図１２】本発明の電子商取引システムへの適用例を示す図。

【符号の説明】

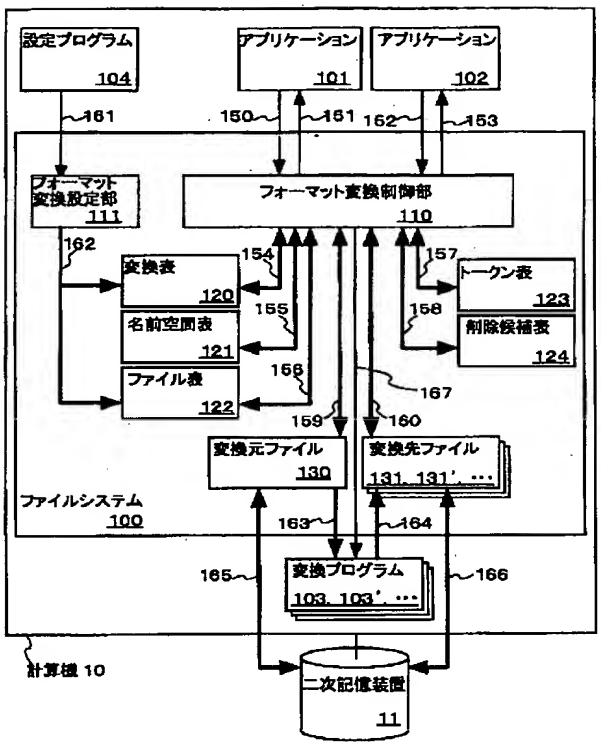
１０：計算機、１１：二次記憶装置、１００：ファイルシステム、１０１、１０２：アプリケーション、１０３、１０３’、…：変換プログラム、１０４：設定プログラム、１１０：フォーマット変換制御部、１１１：フォーマット変換設定部、１２０：変換表、１２１：名前空間表、１２２：ファイル表、１２３：トークン表、１２４：削除候補表、１３０：変換元ファイル、１３１、１３１’、…：変換先ファイル。

【図５】



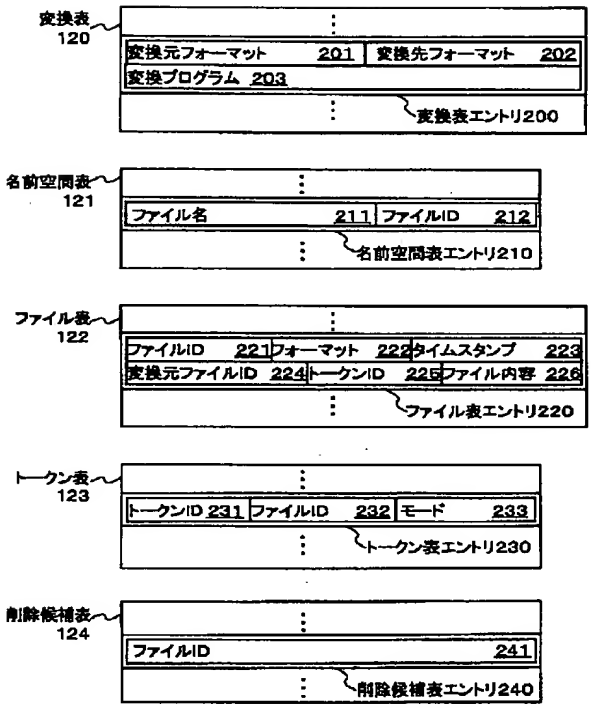
【図１】

図1



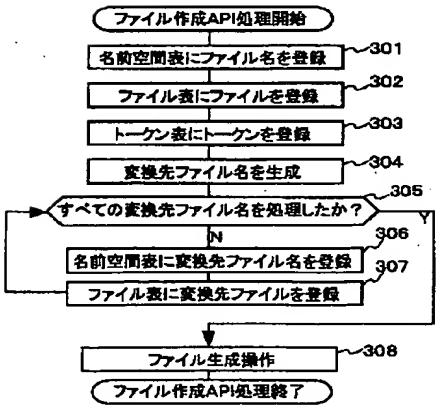
【図２】

図2



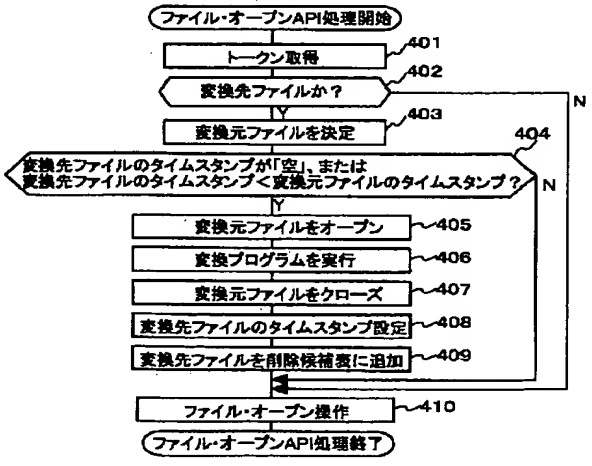
【図３】

図3



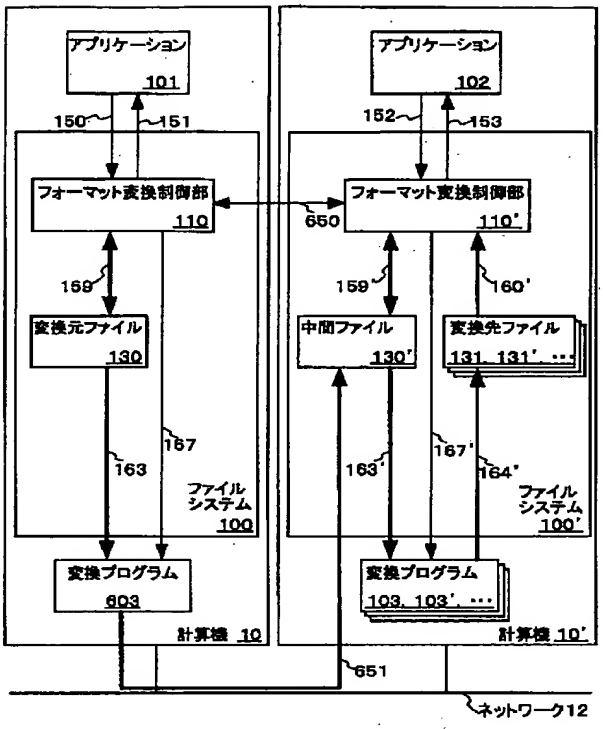
【図４】

図4



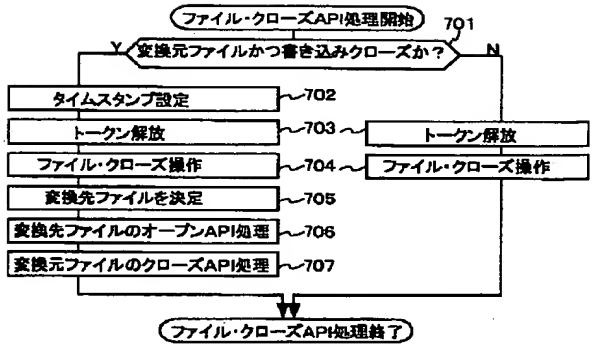
【図６】

図6



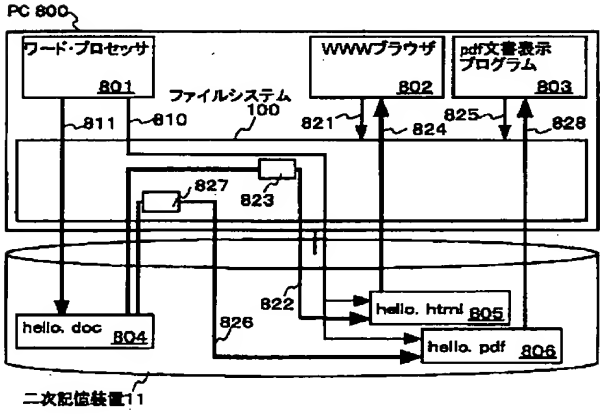
【図７】

図7



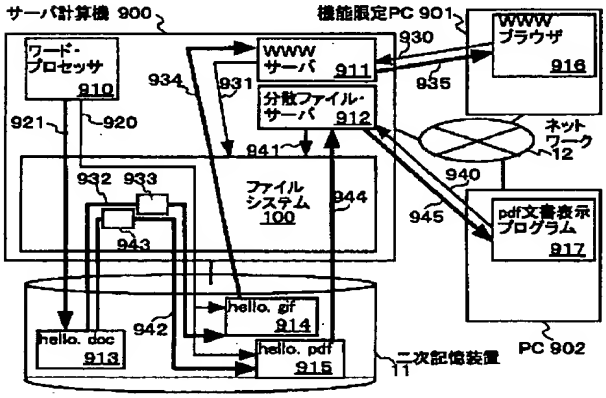
【図８】

図8



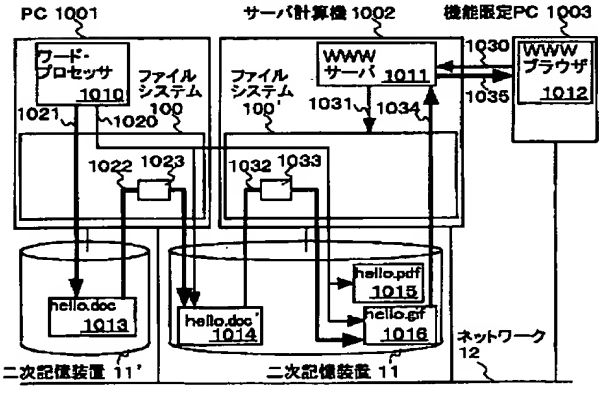
【図９】

図9



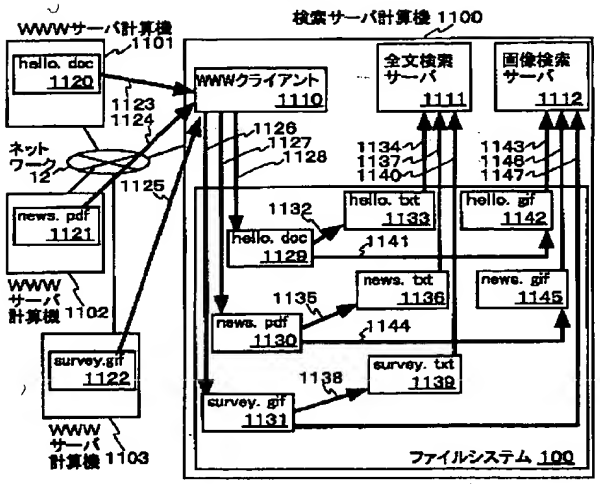
【図１０】

図10



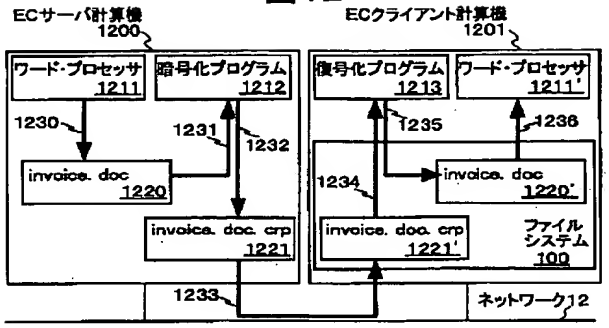
【図１１】

図11



【図１２】

図12



フロントページの続き

(72)発明者　増岡　義政
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者　関　京華
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.